High dimensional discriminant analysis using sparse covariance estimator

by

Jiaxin Zhang

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Statistical Machine Learning

Department of Mathematical and Statistical Sciences
University of Alberta

Abstract

High dimensional classification has drawn massive attention due to its increasing application in genetic diagnosis, image or speech recognition and financial analysis. Traditional methods such as Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA), which are optimal Bayes classifiers under normality assumption, sometimes fail in high dimensional space where the number of variables is considerably greater than the sample size, and thus it is impossible to obtain a good estimation of the covariance matrix by using the conventional empirical estimator. An alternative approach is Naive Bayes which instead assumes all features are independent. Although independence is a critical assumption, it surprisingly does work well in many practical cases. Inspired by the success of Naive Bayes, we aim to find a balance between Naive Bayes and LDA. Hence, it is reasonable to assume only few correlations between features exist in high dimension so that we can take advantage of the sparsity and get a better covariance estimator. The main contribution of this thesis is that we improved the conventional LDA under the sparsity assumption by replacing the empirical covariance estimator with a sparse one. We also review various classification methods specific for high dimensional space. We compared our approach with some of these methods available in R with both simulation and two real data sets and the result showed that our method outperformed many baselines.

Acknowledgments

First I would like to show my deepest appreciation towards my supervisors Dr.Kashlak and Dr.Kong for their irreplaceable help, support, patience and kindness during the past two years. My thesis is based on one of their previous works so I'm sincerely grateful and honored to get involved in this thesis. Thanks to our Department of Mathematical and Statistical Sciences and the University of Alberta for all the resources and support I got. Also I want to thank all classmates who have given me care and support in life, and my graduate student journey is colorful because of them.

Contents

Abstract ii

Acknowledgments iii

1 Introduction 1

2 Methodology 3

2.1 Discriminant Analysis . . . 3

2.1.1 Classification task . . . 3

2.1.2 Bayes Classifier . . . 4

2.1.3 Linear Discriminant Analysis . . . 5

2.1.4 Quadratic Discriminant Analysis . . . 6

2.1.5 Empirical covariance estimator . . . 7

2.1.6 Naive Bayes classifier . . . 9

2.2 Bias variance trade-off in classification . . . 10

2.3 Discriminant Analysis using sparse estimator . . . 13

2.3.1 Sparsity . . . 13

2.3.2 Sparse Covariance Estimation using false positive rate . . . 14

2.3.3 LDAS . . . 15

2.3.4 Thresholding operator . . . 16

3 Previous work 19

3.1 Discriminant analysis . . . 19

3.1.1 SLDA . . . 19

3.1.2 LPD . . . 20

3.2 Feature selection . . . 21

3.2.1 FAIR . . . 22

3.2.2 NSC . . . 23

3.2.3 SCRDA . . . 24

3.3 Projection 25
3.3.1 PCA 25
3.3.2 HDDA 26
3.3.3 Fisher 28
3.3.4 SDA 29
3.3.5 PLDA 30
3.4 Nonlinear classification 31
3.4.1 SVM and Kernels 31
3.4.2 kNN 35
3.4.3 Neural network 35

4 Numerical Results **36**
4.1 Simulation 36
4.1.1 False positive rate and thresholding operator 36
4.1.2 Comparison with other methods 38
4.1.3 Sensitivity to prior 38
4.2 Real data 41
4.2.1 Small Round Blue-Cell Tumor Data 41
4.2.2 Lung Cancer Dataset 44

5 Conclusion **49**

List of Figures

2.1 4 types of sparse covariance matrices. Each pixel represents the correlation and only 20 dimensions are shown for illustration. 10
2.2 Cosines between true slope of discriminant hyperplane for samples from 4 different type of sparse population covariance matrices and the corresponding slopes estimated by LDA and Naive Bayes. Left one shows the simulation result with small sample size while the right one indicates large sample size 11
2.3 Thresholding functions 18

3.1 Distance decomposition figure from [6] 27
3.2 Linearly inseparable example 31
3.3 SVM illustration from [20] 32

4.1 This figure illustrates data generation in first two dimensions. In each dimension the distance between two centroids is 3. 37
4.2 The left graph shows the accuracy of LDAS using different false positive rates with hard thresholding operator, and the right one shows accuracy using four types of thresholding operators with false positive rate $\rho = 0.05$, given data generated with tri-diagonal covariance matrix. The horizontal axis is the distance between two centroids of two groups (ranging from 0 to 2) and the vertical axis is the accuracy of classification. 37
4.3 Relationship between hard and SCAD thresholding 39
4.4 Accuracy curves of 4 type of operators given false positive rate at decreasing level. 39
4.5 Accuracy comparison among different methods. The black line indicates the Bayes error . 40
4.6 Accuracy curves comparison among methods with data generated from three type of sparse covariance matrices 40
4.7 Accuracy curves comparison among different methods in unbalanced cases. 42
4.8 10-fold cross-validation accuracy box-plots of LDAS with different level of false positive rate and thresholding operator compared with other different methods with SRBCT data set 45

4.9 10-fold cross-validation accuracy box-plots of LDAS with different level of false positive rate and thresholding operator compared with other different methods with lung cancer data set 47

List of Tables

4.1 Number of correct classification for each group given 1000 features in lung cancer data set. 48

Chapter 1

Introduction

As one of the most essential tasks of supervised learning, classification has a wide range of application in the real world. For instance, we can predict the gender of a person given information such as height, weight, heart rate or even portrait images. These information are called explanatory variables or features while gender is called the response variable. In this case gender is a categorical factor and it distinguishes classification from regression whose response variable is numeric. The gender of one person can only be either male or female, hence it is called binary classification. There are also cases when the range of response includes more than two categories and it is called multi-classification. For example, as introduced in 4.2.1, we need to make diagnosis for a patient if he is suffering from one of four types of cancers according to his gene expression. In either binary or multi-classification, the sample should be labeled with only one category. This is different from a similar task labeling where one sample can be labeled with either no category or multiple ones.

The reason why we need classification, basically same as the importance of supervised learning, lies in two factors: cost and time. In some cases, it's really expensive or even impossible to obtain some information. One famous example is handwritten digits recognition model by [28]. The cost with respect to both time and money motivated the invention of this recognition model, before which massive transcription work could only be done manually. They tried to teach the computer how to recognize numbers so that the computer can be a substitute with lower cost and liberate people from repetitive and tedious labor. The second factor time refers to prediction tasks. Sometimes there is a lag between the time we need to know the category and the time we can know. A good example from [23] is detecting short lifetime batteries. The lifetime of a battery can never be determined until it fails. However, in order to promise a good quality standard we must try to make sure all the batteries have long lifetime and filter those unqualified out, of course, before they turn out unqualified. Hence, we need to set up a model detecting the short lifetime batteries using features from them.

There are many classification approaches which have achieved huge success such as logistic regression, linear discriminant analysis (LDA) and k nearest neighbor (kNN). LDA is, in theory, the optimal

classifier under normality. However, high dimensionality sets a huge barrier to those conventional approaches, which usually referred to as "curse of dimensionality" [10]. For parametric models, the number of parameters usually explodes dramatically with the increase of dimensionality. For instance, in LDA, the number of parameters is $kp + p(p+1)/2$, which is quadratic with respect to dimension p. Hence, it becomes challenging to build an effective classification model in high dimensional space with limited sample size. There are many previous works that tried to improve LDA by feature selection or projection and many of those assumed the sparsity of correlations among features. Assuming a sparse covariance structure for the features is a common assumption in high dimensional analysis and taking advantage of this assumption enables us to derive better estimation of parameters in the classification model and, as a result, higher accuracy.

In this thesis, we have a review on LDA in section 2 and propose a modification of LDA based on it with respect to parameter estimation by using sparse covariance estimator in section 2.3. We also discuss some previous work in section 3 and the relationship among themselves and with our approach. In section 4, we compare our approach with some previous methods mentioned in section 2 as baselines and both simulation and real high dimensional data set are used.

Chapter 2

Methodology

In this section, we will first give introduction on conventional discriminant analysis and the limitation of these methods when sample size is relatively small in high dimensional space. Motivated by these challenges, we made a reasonable sparsity assumption and improved LDA by using previous work on sparse covariance estimation [24].

2.1 Discriminant Analysis

2.1.1 Classification task

Assume $\mathscr{X}$ is the input space and $\mathscr{Y}$ is the output space which, without loss of generality, can be recoded as $\mathscr{Y} = \{1, 2, ..., K\}$ where K is the number of categories. Usually we have a training data set consisting of n observations or labeled data (X_i, Y_i), $i = 1, ..., n$ and $X_i \in \mathscr{X}$, $Y_i \in \mathscr{Y}$ and it is assumed that each observation is independent. The classification task is to find the best classification function $g : \mathscr{X} \to \mathscr{Y}$ based on the training data (X_i, Y_i).

There are many ways to define what the best function is. The most simple metric is accuracy. Given a new sample $(X, Y) \in \mathscr{X} \times \mathscr{X}$, accuracy is defined as the probability of correct classification, and it can be empirically estimated with another data set, which is usually called test set. The test set must be independent of training set to avoid over-fitting, but is expected to have same distribution as training set and population so that our accuracy estimation is valid.

$$ACC(g) = P(g(X) = Y) \approx \frac{\#\ correct\ \ prediction\ in\ test\ set}{\#\ total\ test\ set}$$

2.1.2 Bayes Classifier

We can also define the best classification function from a different perspective: loss function. The most commonly used loss function for classification task is the zero-one loss:

$$L(y, g(x)) = \begin{cases} 0, g(x) = y \\ 1, g(x) \neq y \end{cases}$$

This loss function can be interpreted as a indicator function of mis-classification as it returns 1 when mis-classification happens. Note that we can also choose different losses for different classes [17] especially in unbalanced cases. Here we only consider consistent loss among all classes. Hence, we look at the expectation of loss function for a classification function g

$$\begin{aligned} W(g|X=x) &= E_Y[L(Y, g(x))] \\ &= L(Y, g(X))P(Y = g(x)|X = x) + L(Y, g(x))P(Y \neq g(x)|X = x) \\ &= P(Y \neq g(x)|X = x) \\ W(g) &= E_{x \in \mathscr{X}}[W(g|X = x)] \\ &= E_{x \in \mathscr{X}}[P(Y \neq g(x)|X = x)] \\ &= P(Y \neq g(X)) \\ &= 1 - ACC(g) \end{aligned}$$

As we can see, the conditional loss function given X is the mis-classification rate of g, and the expected loss function over input space $\mathscr{X}$ is the expectation of the mis-classification rate. Hence, minimizing the expected loss function is equivalent to minimizing the mis-classification rate, or in other words, maximizing the accuracy of classification function g and, if possible, everywhere on input space.

$$\min_{g \in \mathscr{G}} W(g) \equiv \max_{g \in \mathscr{G}} P(Y = g(x)|X = x),\ x \in \mathscr{X}$$

The optimization problem on the right side is to maximize the posterior $P(Y|X)$, hence it is also called Maximize A Posterior(MAP). In theory, the solution to the MAP problem above is called the *Bayes Classifier*, denoted as

$$g^*(x) = arg \max_{g \in \mathscr{G}} P(Y = g(x)|X = x)$$

By Bayes theorem,

$$\begin{aligned} P(Y=k|X=x) &= \frac{P(Y=k,X=x)}{P(X=x)} = \frac{P(X=x|Y=k)P(Y=k)}{P(X=x)} \\ &\propto P(X=x|Y=k)P(Y=k) \\ &= f_k(x)\pi_k \end{aligned}$$

where $f_k(x) = P(X=x|Y=k)$ denotes the conditional density function of x given $Y=k$ and $\pi_k = P(Y=k)$ denotes the prior probability of $Y=k$. Hence, Bayes classifier can also be written as

$$g^\star(x) = arg\max_{k\in\mathscr{Y}} f_k(x)\pi_k$$

Assuming the classification function space is $G = \{g : \mathscr{X} \to \mathscr{Y}\}$, the ideal case is that we can find the "true" function f such that for any $X \in \mathscr{X}$, $f(X) = Y$. However, it is usually impossible to find such function for several reasons. Firstly, information given by input features can be barely as much as that from the output categories. In other words, it is impossible to determine the category based on the observed features with 100% accuracy. Secondly, in all algorithms, many assumptions are made, which narrows the function space G where we looking. It is possible that the true function f is not in this space even if the information we have from the features is enough. For example, in Linear Discriminant Analysis which will be introduced in the following section, we have strong normality and equal variance assumption, which result in a linear classification function. Furthermore, training data we use empirically might mislead us to the function that does not characterize the population best. This happens when the training data is not representative enough due to either noise or sampling bias. Hence, we can only instead try to find the "best" or the "closest" one within function space G based on training data.

2.1.3 Linear Discriminant Analysis

The *Fisher linear discriminant analysis* (LDA) is an extension of Bayes classifier with normality assumption [14] [23], where for each class k, $X \sim N(\mu_k, \Sigma)$, then the density function is

$$f_k(x) = (2\pi)^{-p/2}|\Sigma|^{-1/2}\exp\{-\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)\}$$

Hence, the classification or discriminant function is:

$$\begin{aligned} \delta_{LDA}(x) &= arg\max_{k\in\mathscr{Y}}\{f_k(x)\pi_k\} \\ &= arg\max_{k\in\mathscr{Y}}\{2\log(\pi_k) - ||x-\mu_k||^2_\Sigma)\} \end{aligned}$$

where $||x-\mu_k||_\Sigma^2 = (x-\mu_k)^T\Sigma^{-1}(x-\mu_k)$, which is also called the *Mahalanobis distance*. Assume there are only two classes in output space, $\mathscr{Y} = \{1,2\}$. The Bayes classifier assigns X to class 1 if

$$f_1(X)\pi_1 \geq f_2(X)\pi_2$$

which is equivalent to

$$log\frac{\pi_1}{\pi_2} + (X-\mu)^T\Sigma^{-1}(\mu_1-\mu_2) \geq 0$$

where $\mu = \frac{1}{2}(\mu_1+\mu_2)$. Hence, this is a linear classifier since the discriminant function is a linear function of input feature X. And we can get the loss function ie.e the expectation of mis-classification rate $W(\delta_{LDA})$

$$\begin{aligned}
W(\delta_{LDA}) &= E_{x\in\mathscr{X}}[P(Y \neq \delta_{LDA}(x)|X=x)] \\
&= \int_{x\in\mathscr{X}}[P(Y \neq \delta_{LDA}(x), X=x)]dx \\
&= \pi_1\int_{x\in\mathscr{X}}[P(\delta_{LDA}(x)=2, X=x|Y=1)]dx + \pi_2\int_{x\in\mathscr{X}}[P(\delta_{LDA}(x)=1, X=x|Y=2)]dx \\
&= \pi_1 E_{x\in\mathscr{X}_1}[\mathbf{1}(\delta_{LDA}(x)=2)] + \pi_2 E_{x\in\mathscr{X}_2}[\mathbf{1}(\delta_{LDA}(x)=1)]
\end{aligned}$$

If we assume the balance between two classes i.e. $\pi_1 = \pi_2$, we have $\delta_{LDA}(x) = arg\min_{k\in\mathscr{Y}}\{||x-\mu_k||_\Sigma^2)\}$ and it will assign X as class 1 if $f_1(X) \leq f_2(X)$. Hence we have

$$\begin{aligned}
W(\delta_{LDA}) &= E_{x\in\mathscr{X}_1}[\mathbf{1}(\delta_{LDA}(x)=2)] + E_{x\in\mathscr{X}_2}[\mathbf{1}(\delta_{LDA}(x)=1)] \\
&= \Phi(-\frac{||x-\mu_k||_\Sigma^2}{2})
\end{aligned} \tag{2.1}$$

This is the mis-classification rate of LDA and also the Bayes mis-classification rate under normality assumption, the minimum boundary for any classifier.

2.1.4 Quadratic Discriminant Analysis

In LDA we assume the features in every group follow normal distributions with the same covariance matrix $\Sigma_1 = ... = \Sigma_k = \Sigma$, while in Quadratic Discriminant Analysis (QDA) we loosen this equal variance assumption by assuming different the covariance matrices among groups. Hence the density function of class k is

$$f_k(x) = (2\pi)^{-p/2}|\Sigma_k|^{-1/2}\exp\{-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\}$$

Then the discriminant function of QDA is

$$\begin{aligned}\delta_{QDA}(x) &= arg\max_{k\in\mathscr{Y}} f_k(x)\pi_k \\ &= arg\max_{k\in\mathscr{Y}} \{2\log(\pi_k) - ||x-\mu_k||^2_{\Sigma_k} - \log(|\Sigma_k|)\}\end{aligned}$$

The mis-classification rate for QDA is really complex and since we mainly focus on linear classifier in this thesis, we do not have further analysis on QDA.

2.1.5 Empirical covariance estimator

Either in LDA and QDA, we need to get the estimation the parameters for normal distribution of each group. In section 2.1.2, we showed the equivalence between optimizing loss function and Maximizing A Posterior (MAP). Here we show the MAP estimation of parameters in LDA.

$$\begin{aligned}L(X_i,Y_i) &= f_k(X_i)\pi_k \\ &= (2\pi)^{-p/2}|\Sigma|^{-1/2}\exp\{-\frac{1}{2}(X_i-\mu_k)^T\Sigma^{-1}(X_i-\mu_k)\}\pi_k \\ &\propto |\Sigma|^{-1/2}\exp\{-\frac{1}{2}(X_i-\mu_k)^T\Sigma^{-1}(X_i-\mu_k)\}\pi_k \\ l(\mathscr{X},\mathscr{Y}) &\approx \frac{1}{n}\sum_{i=1}^{n} l(X_i,Y_i) = -\frac{1}{n}\sum_{i=1}^{n}\log(L(X_i,Y_i)) \\ &\propto -\frac{1}{2}\log(|\Sigma|) + \frac{1}{n}\sum_{i=1}^{n}\log(\pi_k) - \frac{1}{2}(X_i-\mu_k)^T\Sigma^{-1}(X_i-\mu_k) \\ \frac{\partial l(\mathscr{X},\mathscr{Y})}{\partial \pi_k} &= \frac{n_k}{n}\frac{1}{\pi_k}, \quad \sum_{k=1}^{K}\pi_k = 1\end{aligned}$$

where n_k denotes the sample size of group k. Hence, the MLE of π_k is $\frac{n_k}{n}$, which is the relative frequency of group k.

$$\begin{aligned}\frac{\partial l(\mathscr{X},\mathscr{Y})}{\partial \mu_k} &= \frac{1}{n}\sum_{Y_i=k} -2\Sigma^{-1}(X_i-\mu_k) \\ &= -2\Sigma^{-1}\left(\frac{\sum_{Y_i=k}X_i}{n} - \mu_k\right)\end{aligned}$$

Setting this derivative to zero, we can get the MLE of $\mu_k = \frac{\sum_{Y_i=k} X_i}{n}$, which is the sample mean of group k.

$$\begin{aligned} l(\mathscr{X},\mathscr{Y}) &\propto -\frac{1}{2}\log(|\Sigma|) + \frac{1}{n}\sum_{i=1}^{n} -\frac{1}{2}(X_i-\mu_k)^T\Sigma^{-1}(X_i-\mu_k) \\ &= \frac{1}{2}\log(|\Sigma^{-1}|) + \frac{1}{n}\sum_{i=1}^{n} -\frac{1}{2}Trace(\Sigma^{-1}(X_i-\mu_k)(X_i-\mu_k)^T) \\ &= \frac{1}{2}\log(|\Sigma^{-1}|) - \frac{1}{2}Trace(\Sigma^{-1}\frac{1}{n}\sum_{i=1}^{n}(X_i-\mu_k)(X_i-\mu_k)^T) \\ \frac{\partial l(\mathscr{X},\mathscr{Y})}{\partial \Sigma^{-1}} &= \frac{1}{2}\Sigma - \frac{1}{2n}\sum_{i=1}^{n}(X_i-\mu_k)(X_i-\mu_k)^T \end{aligned}$$

By setting this derivative to zero, we derive the MLE of Σ

$$\hat{\Sigma}^{MLE} = \frac{1}{n}\sum_{i=1}^{n}(X_i-\mu_k)(X_i-\mu_k)^T$$

which is the sample covariance or *empirical covariance estimator*. Since $E[\hat{\Sigma}^{MLE}] = \frac{n}{n-k}\Sigma$, we can also scale the MLE to obtain an unbiased estimator

$$\hat{\Sigma}^{unbias} = \frac{n-k}{n}\hat{\Sigma}^{MLE} = \frac{1}{n-k}\sum_{i=1}^{n}(X_i-\mu_k)(X_i-\mu_k)^T$$

In QDA, we have same MLE for π_k and μ_k and similar estimator for covariance matrix Σ_k

$$\hat{\Sigma}_k^{MLE} = \frac{1}{n_k}\sum_{i=1}^{n_k}(X_i-\mu_k)(X_i-\mu_k)^T$$

In theory, LDA and QDA are the Bayes classifier under their normality assumptions respectively. Empirically, we build our classifier by estimating the normality parameters used in the classification function. Hence, there are two sources where possible problems that might limit the practical performance of these methods : normality and parameter estimation. For normality assumption, this is the common assumption we have in most statistical analysis because of its good properties. We also have approaches for checking normality [26]. For parameter estimation, the number of parameters in covariance matrix under the normality assumption in LDA is $p(p+1)/2$ and in QDA this number should be multiplied by K. In order to promise accurate estimations, the amount of training data we need grows at least quadratically with the increase of dimensions while in high-dimensional classification, we usually have training data with relatively limited sample size. The most direct impact is the breakdown of classification rule in either LDA or

QDA due to the singularity of covariance estimator.

$$rank(\hat{\Sigma}) \leq \sum_{i=1}^{n} rank((X_i - \mu_k)(X_i - \mu_k)^T) = n$$

The `lda` model in `MASS` package [37] just ignore the potential singularity issue by projecting the original data into some subspace if the covariance estimator is not full ranked. More specifically, after normalizing the data, we get the singular value decomposition of the covariance estimator.

$$\hat{\Sigma} = V^T \Lambda V = (V_r, V_{p-r}) \begin{pmatrix} \Lambda_r & 0 \\ 0 & 0 \end{pmatrix} (V_r, V_{p-r})^T = V_r \Lambda_r V_r^T$$

where Λ_r is the diagonal matrix consisting of non-zero singular values of $\hat{\Sigma}$. Then we project the original data in to a small r-dimensions space $X' = \Lambda_r^{-1/2} V_r^T X$ and use the Euclidean distance directly in classification function, which is also the Mahalanobis distance since the covariance matrix after projection is identity matrix.

$$Cov(X') = \Lambda_r^{-1/2} V_r^T \hat{\Sigma} (\Lambda_r^{-1/2} V_r^T)^T = I_r$$

In order to promise the stability, singular values smaller than the tolerance (1e-04 by default) should be regarded as zero. Hence, the subspace dimension r might be less than the rank of empirical estimator. This idea makes sense when we have a good estimation of covariance matrix and we can find a subspace consisting of dimensions with large variation and dense information. However, based on the limited sample size, the large variation of covariance estimator make itself unreliable and as a result, projection based on the this bad estimation is likely to leave the informative features or dimensions out.

2.1.6 Naive Bayes classifier

Another available solution to addressing the singularity issue is Naive Bayes, where not only normality but also independence of features are assumed at the same time [36]. In other words, it can be considered as a LDA model with the constraint that the covariance matrix is diagonal.

$$\begin{aligned} f_k(x) &= \prod_{j=1}^{p} p(x_j | y = k) \\ &= \prod_{j=1}^{p} \frac{1}{\sqrt{2\pi\sigma_{k,j}^2}} \exp(-\frac{(x_j - \mu_{k,j})^2}{\sigma_{k,j}^2}) \\ &= (2\pi)^{-p/2} |D|^{-1/2} \exp\{-\frac{1}{2}(x - \mu_k)^T D^{-1} (x - \mu_k)\} \end{aligned}$$

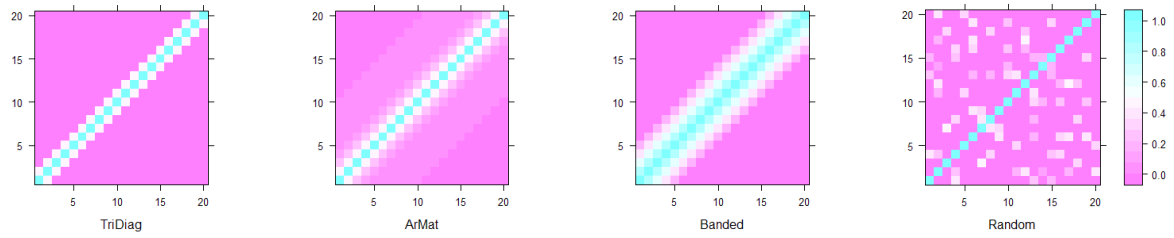

Figure 2.1: 4 types of sparse covariance matrices. Each pixel represents the correlation and only 20 dimensions are shown for illustration.

where D is the diagonal covariance matrix with only diagonal entries $\sigma_{k,j}$ non-zero. Hence, the number of parameters in covariance matrix drops significantly from $p(p+1)/2$ to p and this diagonal covariance matrix not singular as it is in LDA using empirical estimation.

We may doubt if the independence assumption is reasonable in practice or, in other words, if a diagonal matrix is a good estimator of covariance matrix. In high dimensional space where there are a large amount of features, we expect most of them are independent while some of them might still be highly correlated with one another. Although the independence assumption makes it possible to get a non-singular covariance matrix, some off-diagonal entries are ignored and set as zero directly. Hence, compared with empirical estimator, the diagonal one we use in Naive Bayes is obviously a biased estimator. However, the variance of the Naive Bayes estimator is much smaller than the empirical estimator due to many fewer parameters to estimate.

2.2 Bias variance trade-off in classification

There is a trade-off between bias and variance and we tried a simulation in order to verify the trade-off. Recall that the discriminant functions for either LDA or Naive Bayes is a linear function with respect to x, it can be interpreted as defining a hyperplane $\{x : x^T\beta + \beta_0 = 0\}$, which splits the input space into two parts corresponding to two classes. The slope defined by LDA discriminant function is $\beta_{LDA} = \Sigma^{-1}(\mu_1 - \mu_2)$ while the slope for Naive Bayes is $\beta_{NB} = D^{-1}(\mu_1 - \mu_2)$, and the intercept $\beta_0 = log\frac{\pi_1}{\pi_2} - \frac{1}{2}(\mu_1 + \mu_2)^T\Sigma^{-1}(\mu_1 - \mu_2)$. To simplify the simulation, we focus on balanced case i.e. $log\frac{\pi_1}{\pi_2} = 0$ and the data after normalizing should have zero means i.e. $(\mu_1 + \mu_2) = 0$. Hence we can ignore the intercept and focus on the estimation of slopes. We generated 20 sample (10 for each class) in $\mathscr{R}^{100}$ following multi-variate Gaussian distribution with $\mu_1 = -1, \mu_2 = 1$ and 4 types of sparse covariance matrices [24]: Tri-Diagonal, Auto-Regressive, Banded and Random Sparse Matrix as shown in 2.1. We compare how close the estimations from LDA and Naive Bayes are to the ground truth. Since slope with different scales might determine the same hyperplane, we use the cosine of slopes as the measurement of similarity or closeness.

The left box-plot in Figure 2.2 shows result from 1000 replicates. As we expect, the variation of

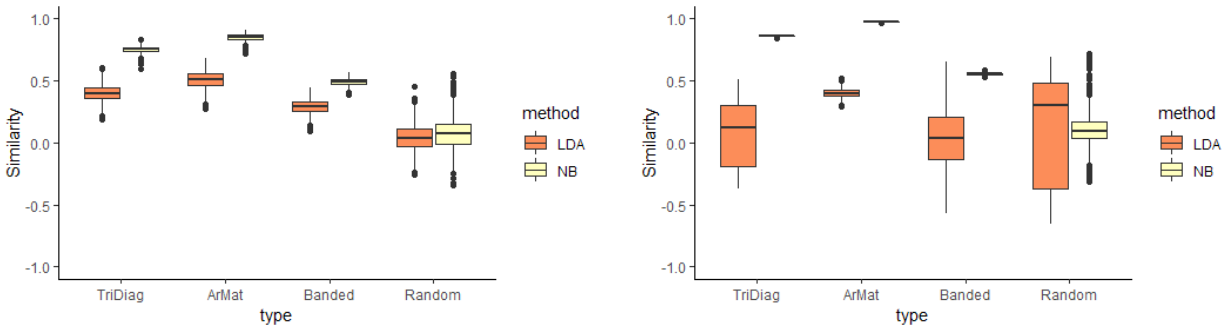

Figure 2.2: Cosines between true slope of discriminant hyperplane for samples from 4 different type of sparse population covariance matrices and the corresponding slopes estimated by LDA and Naive Bayes. Left one shows the simulation result with small sample size while the right one indicates large sample size

Naive Bayes estimation is much lower than LDA which indicates the stability of Naive Bayes. However, it is surprising that Naive Bayes also achieved higher similarity on average. In theory we expect that LDA should achieve unbiased estimation and, as a result, a higher similarity, but it turned out that the "unbiased" property is not as expected. One reason behind can be the inverse of empirical estimator. With limited sample size, empirical estimator holds large variation and low rank. Thus, taking inverse of this estimator leads to large bias and exaggerates the variation, which invalidates the "unbiased" property and results in bad estimation.

We also did a simulation with large sample size where we generated 200 sample (100 for each class) and the right box-plot in Figure 2.2 shows result. Still, in the first three cases, Naive Bayes outperformed LDA, which attributes to the sparsity of covariance matrices. However, the bias variance trade-off is more clear for a random covariance matrix. Large sample size allows empirical estimator to be more accurate than diagonal one on average and, as a result, a higher similarity and less bias of LDA than Naive Bayes.

On balance, for either small or large sample size, Naive Bayes outperforms LDA for all three sparse correlation cases with respect to both bias and variance, which might be attributed to the consistency between independence assumption in Naive Bayes and the sparsity prior we set in simulation. For random covariance matrices, there is a trade-off between variance and bias when sample size is large, Naive Bayes still achieves higher similarity with limited sample size.

Another previous work [17] also tried to explain effect of bias and variance in classification but from a more straight forward perspective we discussed in section 2.1.2, classification error rate or equivalently accuracy. Instead of considering the slope of the hyperplane, they focus on the posterior probability $p(y|x)$ directly. Assume there are only two balanced classes $\mathscr{Y} = \{0,1\}$ and $P(Y=1|X=x) = 1-P(Y=0|X=x) \doteq f(x)$. Recall that the Bayes classifier is defined as

$$g^*(x) = arg\max_{g\in\mathscr{G}} P(Y=g(x)|X=x),$$

then the Bayes classifier becomes $g^*(x) = 1(f(x) \geq 1/2)$. The classification error rate becomes

$$\begin{aligned} P(y_B \neq y|x) &= \begin{cases} P(y=1|x) \; if \; y_B = 0 \; i.e. \; P(y=0|x) > 1/2 \\ P(y=0|x) \; if \; y_B = 1 \; i.e. \; P(y=1|x) > 1/2 \end{cases} \\ &= 1(f(x) < 1/2) f(x) + 1(f(x) \geq 1/2)(1 - f(x)) \\ &= min\{f(x), 1 - f(x)\} \end{aligned}$$

Given the training set T, the estimation of $f(x)$ is denoted as $\hat{f}(x|T)$. Denote the prediction by $\hat{f}(x|T)$ as $\hat{y}$, we have error rate for $\hat{y}$ as follow

$$\begin{aligned} P(\hat{y} \neq y|x) &= \begin{cases} \min\{f(x), 1 - f(x)\} \; if \; \hat{y} = y_B \\ \max\{f(x), 1 - f(x)\} \; if \; \hat{y} \neq y_B \end{cases} \\ &= |f(x) - (1 - f(x))| P(\hat{y} \neq y_B) + \min\{f(x), 1 - f(x)\} \\ &= |2f(x) - 1| P(\hat{y} \neq y_B) + P(y_B \neq y) \end{aligned}$$

The second term $P(y_B \neq y)$ is the irreducible error rate and what we can do is to minimize $P(\hat{y} \neq y_B)$, which is called "boundary error", as much as possible. At each input point x, $\hat{f}(x|T)$ is a function of T and can be regarded as a random variable due to the randomness of T. Hence, without any assumption on the distribution of $\hat{f}(x|T)$, we can calculate the "boundary error"

$$P(\hat{y} \neq y_B|x) = 1(f(x) < 1/2) \int_{1/2}^{+\infty} p(\hat{f}) d\hat{f} + 1(f(x) \geq 1/2) \int_{-\infty}^{1/2} p(\hat{f}) d\hat{f}$$

In order to have further analysis on this boundary error, they assumed $\hat{f}(x|T)$ follows a normal distribution at point x, with mean and variance defined as usual

$$\begin{aligned} E[\hat{f}] &= \int_{-\infty}^{+\infty} \hat{f} p(\hat{f}) d\hat{f} \\ var(\hat{f}) &= \int_{-\infty}^{+\infty} (\hat{f} - E[\hat{f}])^2 p(\hat{f}) d\hat{f} \end{aligned}$$

Under the normality assumption, the boundary error rate becomes

$$P(\hat{y} \neq y_B|x) = \phi\left[\frac{sign(f - 1/2)(E[\hat{f}] - 1/2)}{\sqrt{var(\hat{f})}}\right]$$

where $\phi(z)$ is the right tail probability of the standard normal distribution. The opposite number of nu-

merator in this function is defined as "boundary bias".[17]

$$b(f,\hat{f}) = -sign(f-1/2)(E[\hat{f}]-1/2)$$

Although the normality assumption may be not valid, we can still get some inspiration from this approximation. As observed, the bias is defined differently and the bias and variance do not work in an additive way as in regression any longer. In order to promise a effective classifier with classification error less than 50%, first we need to make sure the boundary bias is negative, which means $(f-1/2)(E[\hat{f}]-1/2)$ is positive or in other words, the expected prediction $\hat{y}$ should be consistent with y_B. With boundary bias being negative, we can reduce the variance or increase $|E[\hat{f}]-1/2|$, either of which leads to a more confident prediction.

The conclusion above explained why Naive Bayes can achieve better performance to some extend. With simple independence assumption and less parameters to estimate in the model, Naive Bayes approach results in $\hat{f}_{NB}$ with considerably less variance than $\hat{f}_{LDA}$. However, Neither Naive Bayes or LDA is perfect enough. With large sample size, LDA is likely to achieve estimation with low variance and less bias to Bayes classifier while Naive Bayes approach can not diminish the bias resulting from independence assumption no matter whether the sample size is large or not. Hence, it inspired us to find if there is a balance between these two methods which can achieve both acceptable variance and less bias. One approach is to recover those off-diagonal entries which are significantly different from zero and remain the stability of estimation at the same time, so that we can have a better estimation of covariance matrix than a simple diagonal one and expect to have less bias in average. Some works on covariance estimation will be introduced in the next section.

2.3 Discriminant Analysis using sparse estimator

2.3.1 Sparsity

Recall that in Naive Bayes, the independence assumption makes all the off diagonal entries in covariance matrix zero, our aim is to recover those non-zeros entries. A class of sparse matrices enables us to have further theoretical analysis with its desired properties. The class is defined as below

$$\mathscr{U}(k,\sigma) = \{\Sigma \in \mathscr{R}^{p\times p} : max_{j=1,\ldots,p} \sum_{i=1}^{p} \mathbb{1}[\sigma_{i,j} \neq 0] \leq k,\ \sigma = inf_{i<j}\{|\sigma_{i,j}| > 0\}\}$$

In other words, it is assumed that for each column in a sparse covariance matrix, the number of non-zero entries does not exceed k and it is also true for each row since covariance matrix is symmetric. This assumption can be interpreted as that each variable is correlated to at most k variables and is independent

with all the other variables. Also, nonzero entries are bounded away from zero by σ. The Sparsity assumption is serving as a prior knowledge on covariance matrix. It is more specific and targeted in high dimensional space compared with LDA where no constraint is imposed on covariance matrix, and more flexible than Naive Bayes or Independence rule. Under the sparsity assumption, we can do further analysis on covariance matrix estimation and improve LDA with a better estimator.

2.3.2 Sparse Covariance Estimation using false positive rate

The estimation of covariance matrices in high dimensional space is a focused research area since its wide application in not only LDA, but also principle component analysis, regression and other areas of statistical inference. Sparse covariance estimation is a balance between empirical estimator and independence rule, where we have sparsity assumption on covariance matrix. There are mainly three classes of sparse estimators: shrinkage estimators, which shrink estimated eigenvalues, eigen-vectors or the matrix itself towards some desired target; regularized estimator with lasso-penalties; thresholding estimator, which threshold the covariances in the covariance matrix to zero entry-wise. In this thesis, we use one thresholding sparse estimator proposed by [24], which is not only computationally efficient but also more interpretable.

Many previous works share the idea of thresholding. A generalized thresholding operator, as defined in [30], is $s_\lambda(\cdot) : \mathscr{R} \to \mathscr{R}$ such that

$$|s_\lambda(z)| \leq z,\ s_\lambda(z) = 0\ for |z| \leq \lambda,\ |s_\lambda(z) - z| \leq \lambda$$

which will apply element-wise to a matrix. There are many types of thresholding operators which will be discussed later in section 2.3.4. In general, the key to thresholding is the choice of parameter λ. Since this parameter λ is not interpretable enough, many previous works such as [2][3][30][7] only choose the "best" λ via cross validation, which is usually time consuming. However, the approach we proposed uses *false positive rate* to guide the choice of λ, which is not only time efficient but also more interpretable.

For an estimator $\hat{\Sigma} \in \mathscr{R}^{p\times p}$, the *false positive rate* is

$$\rho(\hat{\Sigma}) = \frac{\#\{\hat{\sigma}_{i,j} \neq 0 | \sigma_{i,j} = 0, i < j\}}{p(p-1)/2}$$

where $\sigma_{i,j}$ is the i,j th entry of the true covariance matrix Σ and $\hat{\sigma}_{i,j}$ is the i,jth entry of the estimator $\hat{\Sigma}$. The numerator is the number of false positive entries and the denominator is total number of off diagonal entries. Naive Bayes is an extreme case with false positive rate zero since all the off diagonal entries are estimated zero under independence assumption.

Based on the sparsity assumption and thresholding properties in appendix: lemma 5.0.1 and theorem 5.0.2, we can find the sparsest threshold estimator approximately achieving a desired false positive rate. Given a false positive rate ρ, empirical covariance estimator $\hat{\Sigma}^{emp}$ and a thresholding operator s_λ, the steps

of our approach is as below.

Inputs : $\hat{\Sigma}^{emp}, s_\lambda, \rho$

Step 0 : Set $\hat{\Sigma}_0^{sp} = (\hat{\Sigma}^{diag})^{-\frac{1}{2}} \hat{\Sigma}^{emp} (\hat{\Sigma}^{diag})^{-\frac{1}{2}}$ to be the empirical estimator normalized to have a diagonal of ones.

Step 1 : Find $\eta = 2^a \rho \ st. \ \eta \in (0.5, 1), a \in \mathscr{Z}^+$. Find the η quantile M_η of all off diagonal entries $\{\sigma_{i,j}, \ i < j\}$ in $\hat{\Sigma}_0^{sp}$.

Step 2 : Calculate $r_\rho = \frac{1}{2^a} ||s_{M_\eta}(\hat{\Sigma}_0^{sp}) - \hat{\Sigma}_0^{sp}||_\infty$.

Step 3 : Find the largest $\lambda \ st. \ ||s_\lambda(\hat{\Sigma}_0^{sp}) - \hat{\Sigma}_0^{sp}||_\infty \leq r_\rho$.

Step 4 : Return $\hat{\Sigma}^{sp} = (\hat{\Sigma}^{diag})^{\frac{1}{2}} s_\lambda(\hat{\Sigma}_0^{sp}) (\hat{\Sigma}^{diag})^{\frac{1}{2}}$

Output : $\hat{\Sigma}^{sp}$

Transformation in Step 0 allows us to do thresholding on correlation matrix rather than covariance matrix and in step 4 we do the transformation backwards to the original scale. In step 1, we get the threshold for a false positive rate $\eta \geq 0.5$ as mentioned in lemma 5.0.1 and then step 2 follows the conclusion in 5.0.2 which enables us to get the approximate distance for a threshold estimator achieving desired false positive rate ρ to the true correlation matrix which is estimated by empirical estimator $\hat{\Sigma}_0^{sp}$. In Step 3 we search the sparest estimator that falls within this distance.

2.3.3 LDAS

As we discussed in section 2.1.5, one issue for LDA is the estimation of covariance matrix while section 2.3.2 provides an effective and efficient solution to this issue. Hence, it is natural to combine these two ideas by replacing the empirical estimator in LDA with the sparse estimator and we name it LDAS. Given an labeled training data set consisting of n observations (X_i, Y_i), $i = 1, ..., n$ and $X_i \in \mathscr{X}$, $Y_i \in \mathscr{Y}$, we firstly get the sparse covariance estimate $\hat{\Sigma} = \Sigma_\rho^{sp}$ with fixed false positive rate $\rho \in [0, 1]$. Then use the LDA classification function with the new sparse covariance estimator.

$$\delta_{LDAS}(x) = arg\max_{k \in \mathscr{Y}} \ 2log(\pi_k) - ||x - \mu_k||^2_{\Sigma_\rho^{sp}})$$

This discriminant function is same as LDA when the false positive rate is set as 1 since no thresholding will be imposed on covariance matrix entries, and equivalent to Naive Bayes when false positive rate is 0 as all the off-diagonal elements will be shrunken as zero. Hence, we conclude that this method is a balance between LDA and Naive Bayes, where we can choose how many off-diagonal elements in covariance

matrix to keep by adjusting the false positive rate from 0 to 1. With the flexibility of choice on false positive rate, LDAS is at least competitive to its parents LDA and Naive Bayes and hopefully have better performance in some cases when neither LDA or Naive Bayes works.

2.3.4 Thresholding operator

The sparsity assumption is necessary for the uniqueness of solution and the key step in estimation sparse estimator is to find the sparsest estimation by thresholding. [30] concluded a generalized form of thresholding $s_\lambda(\cdot) : \mathscr{R} \to \mathscr{R}$ satisfying the following conditions

1. $|s_\lambda(z)| \leq |z|$;
2. $s_\lambda(z) = 0 \; for \; |z| \leq \lambda$;
3. $|s_\lambda(z) - z| \leq \lambda$

The first condition shows that the thresholding result should be no greater than the original input. The second one explains how the thresholding functions for input values smaller than thresholds. The third one indicates that the thresholding will only shrink the input in a limited scale so that large input will not be effected much after thresholding. If we combine the first and the third one, we will have $0 \leq z - s_\lambda(z) \leq \lambda$. It is also natural to have $s_\lambda(z) = sign(z)s_\lambda(|z|)$ which means the thresholding keeps the original sign of input, but this condition is not strictly necessary.

The simplest thresholding is hard thresholding rule:

$$s_\lambda^{Hard}(z) = z \cdot \mathbb{1}(|z| > \lambda)$$

The hard thresholding rule can filters out values that are greater than threshold λ and only shrinks small values to zeros. Obviously, it satisfies all three conditions above and it only has effect on small values. One disadvantage is that the thresholding function is not continuous.

Soft thresholding results from lasso penalty and the thresholding function is:

$$s_\lambda^{Soft}(z) = sign(z) \cdot (|z| - \lambda)_+$$

In a nutshell, soft thresholding shrinks all the values towards zero by at most λ. Values less than λ will be shrunk as zero, which is consistent with the second condition in definition of generalized thresholding. For values great than λ, the shrinkage by soft thresholding is λ,the maximum amount the third condition allows, which, however, also makes the thresholding function continuous.

The Smoothly clipped absolute deviation (SCAD) penalty proposed by [15] finds a balance between hard and soft thresholding. Its thresholding function is continuous and the shrinkage will decease as z

increases and no shrinkage after a certain value. Besides λ, there is another unknown parameter a ($a > 2$), which can be either the suggested value ($a = 3.7$) in [15] or determined be cross validation along with tuning λ.

$$s_\lambda^{SCAD}(z) = \begin{cases} sign(z) \cdot (|z| - \lambda)_+, \ for \ |z| \leq 2\lambda \\ \dfrac{(a-1)z - sign(z)a\lambda}{a-2}, \ for \ 2\lambda < |z| \leq a\lambda \\ z, \ |z| > a\lambda \end{cases}$$

The SCAD is exactly same as soft thresholding for $|z| \leq 2\lambda$ and hard thresholding for $|z| > a\lambda$. Hence the key of switching from soft to hard thresholding is the second part for $2\lambda < |z| \leq a\lambda$. It can be proved continuous in $z \in \mathscr{R}$ as follow:

$$lim_{z \to 2\lambda^+} s_\lambda^{SCAD}(z) = \frac{(a-1)2\lambda - a\lambda}{a-2} = \lambda = lim_{z \to 2\lambda^-} s_\lambda^{SCAD}(z)$$
$$lim_{z \to a\lambda^-} s_\lambda^{SCAD}(z) = \frac{(a-1)a\lambda - a\lambda}{a-2} = a\lambda = lim_{z \to a\lambda^+} s_\lambda^{SCAD}(z)$$

The last type of thresholding is adaptive lasso from [39]. The "adaptive" can be reflected by the shrinkage term $\lambda^{\eta+1}|z|^{-\eta} = \lambda \cdot (\frac{\lambda}{z})^\eta$. Compared with soft thresholding rule whose shrinkage is λ, adaptive lasso imposes smaller penalty on large values ($z \geq \lambda$), and the larger the value is the smaller the penalty will be, hence, approaching hard thresholding when $z \to \infty$.

$$s_\lambda^{Adpt}(z) = sign(z) \cdot (|z| - \lambda^{\eta+1}|z|^{-\eta})_+$$

From Figure 2.3 we can compare the relationship and the difference among these four types of thresholding functions. The shade area represents where any possible thresholding function defined as 2.3.4 can be. As shown in the graph, hard and soft thresholding function is the lower and upper bound of the shade area respectively. Both SCAD and adaptive lasso methods are trying to switch from soft to hard so that less bias will be imposed on input above the threshold.

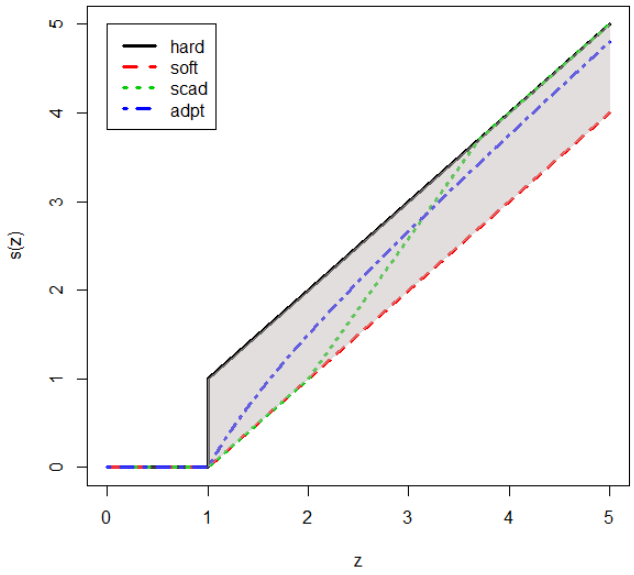

Figure 2.3: Thresholding functions

Chapter 3

Previous work

3.1 Discriminant analysis

Although LDA is the Bayes classifier under the normality assumption, we cannot promise a good empirical performance without good estimation of parameters, which is the key to determine a classification function. There are many previous works trying to improve LDA and most of them took advantage of the properties in high dimensional space such as sparsity to obtain better estimates.

3.1.1 SLDA

The Sparse Linear Discriminant Analysis (SLDA) proposed by [31] was based on the sparsity assumption of both covariance matrix $\Sigma = \{\sigma_{i,j}\}$ and the difference between group means $d = \mu_1 - \mu_2$. The idea is to find better estimation of d and Σ used in LDA with these sparsity assumption in high dimensional space. The sparse estimator they used for covariance matrix was proposed in [2], where the measure of sparsity is defined as below

$$C_h = max_{i \leq p} \sum_{j=1}^{p} |\sigma_{i,j}|^h,\ h \in [0,1)$$

When $h = 0$, $|\sigma_{i,j}|^0 = 1$ for $\sigma_{i,j} \neq 0$,and $|\sigma_{i,j}|^0 = 0$ for $\sigma_{i,j} = 0$. Hence, it is still consistent with the measure we used in defining the sparsity space in section 2.3.1. Then the covariance estimate can be obtained by imposing hard thresholding on empirical estimator.

$$\widetilde{\Sigma} = s_{\lambda}^{hard}(\hat{\Sigma}^{emp}) = \{\sigma_{i,j} \mathbb{1}(|\sigma_{i,j}| > \lambda_1)\},\ \lambda_1 = M_1 \frac{\sqrt{\log p}}{\sqrt{n}}$$

where M_1 is a positive constant and only off diagonal elements will be thresholded. They also defined the sparsity measure on d:

$$D_g = \sum_{i=1}^{p} |d_i|^{2g},\ g \in [0,1)$$

where d_i is the ith component of d, g is a constant not depending on p. Similarly, this measure can be regarded as a count of non-zero elements when $g = 0$. Hence, we can also obtain a sparse estimate of d by thresholding.

$$\widetilde{\mu} = s_\lambda^{hard}(\hat{\mu}) = \{\mu_i \mathbb{1}(|\mu_i| > \lambda_2)\},\ \lambda_2 = M_2(\frac{\log p}{n})^\alpha$$

where M_2 is a positive constant and $\alpha \in (0, 1/2)$.

3.1.2 LPD

SLDA takes advantage of sparsity of both d and Σ and get sparse estimates on them separately. However, recall that for binary classification task, we will classify an input X as class 1 if

$$log\frac{\pi_1}{\pi_2} + (X-\mu)^T\Sigma^{-1}d \geq 0$$

From the equation we can conclude that in order to determine the classification function, it is not necessary to estimate both Σ and μ exactly. Instead, we only need the the product of the inverse matrix or precision matrix $\Omega = \Sigma^{-1}$ and d, and this is main idea of Linear Programming Discriminant (LDP) [8]. Compared with estimating Σ and μ separately, the number of parameters that need estimating for product $\beta = \Omega d$ drops significantly from $p(p-1)/2 + p$ to p.

Ideally we want to find β such that $\Sigma\beta = d$. Since in high dimensional space the sample size is likely to be less than the dimensions, the empirical estimate of covariance matrix is singular. This results in infinite solutions for β, in other words, infinite elements in feasible set $\{\beta :\ \hat{\Sigma}^{emp}\beta = \hat{d}\}$. In order to make the solution unique, naturally, LPD assumes that β is sparse which is similar to the sparsity assumption of Σ and μ in SLDA, and seeks the most sparse solution from within the feasible set. Since the estimation of neither Σ or μ can be accurate, we loosen the feasible set to $\{\beta :\ |\hat{\Sigma}^{emp}\beta - \hat{d}|_\infty \leq \lambda_n\}$, where λ_n is a tuning parameter that measures the acceptable deviation. Hence, the classification rule will change into

$$log\frac{\pi_1}{\pi_2} + (X-\mu)^T\hat{\beta} \geq 0$$

where the estimation $\hat{\beta}$ can be obtained by solving the following optimization problem:

$$\hat{\beta} = arg\min_{\beta \in \mathscr{R}^p}\{|\beta|_k\ subject\ to\ |\hat{\Sigma}^{emp}\beta - \hat{d}|_\infty \leq \lambda_n\}$$

Both SLDA and LDP aim to find the sparsest estimation of parameters in classification function by hard thresholding. In SLDA the tuning parameter is thresholds while in LDP the tuning parameter is λ_n, which can be regarded as tolerance for the difference between new estimation and empirical one. These tuning parameters might differ among data in different scale, so there is no general conclusion on selection of tuning parameters and the the only way to determine is cross-validation. Hence, although the drop of the amount of tuning parameters makes LDP more computationally efficient compared with SLDA, cross-validation is still unavoidable, which is the bottle neck of computation efficiency.

The difference of assumption also indicates that they focus on different situations. [8] explained the relationship between assumptions of LDP and SLDA. A vector β is called $s-sparse$ if it has at most s nonzero entries. Similarly, a matrix Σ is called $s-sparse$ if each row(column) has at most s nonzero entries. Then we have the following conclusion:

Remark. *from [8] If $d \in \mathscr{R}^p$ is s_1-sparse and $\Omega \in \mathscr{R}^{p\times p}$ is s_2-sparse, then Ωd is at most $s_1 s_2$-sparse.*

Proof. Without loss of generality, we assume the first s_1 entries of d are nonzero. Then we can write $d = (d_1, ..., d_{s_1}, 0)^T$, $\Omega = \{\Omega_1, \Omega_2, ..., \Omega_p\}$, where $\Omega_i \in \mathscr{R}^p$. Then the product is $\Omega d = \sum_{i=1}^{s_1} \Omega_i d_i$. Since Ω_i is s_2-sparse, $\sum_{i=1}^{s_1} \Omega_i d_i$ can be at most $s_1 s_2$-sparse. □

Remark 3.1.2 shows that the sparsity assumption in SLDA is a sufficient condition for that in LDP when $s_1 s_2$ is smaller than the dimension p. On the other hand, there are also cases where neither Ω nor d is sparse but Ωd is. For example, assume that Ω is an orthogonal p-sparse matrix and d is one p-sparse column of Ω, then $\Omega d = (0, ..., 1, ...0)$ which is 1-sparse. Hence, the sparsity on Ωd is, to some extend, more flexible than assuming both Ω and d are sparse.

The sparsity assumptions in these two methods are different from our sparsity assumption on covariance matrix. Obviously, the assumption in SLDA is a special case for our assumption. However, the assumption in LPD is somehow invalid if the features are screened before feeding into the classifier. LPD assumes the β is sparse which is the slope the hyperplane determined by the classification rule. Hence, This sparsity indicates that there are uninformative features and basically LPD is somehow equivalent to feature selection. Our model, instead, only assumes the sparsity on correlations among features and we will take the information from all features. Hence, one underlying assumption is that the features we have are informative, which is totally different from LPD.

3.2 Feature selection

Feature selection is another solution to high dimensional classification. Instead of searching for better estimation of parameters, feature selection reduces the dimension first and hence, the number of parameters to be estimated. The sparsity of d is also a key assumption in feature selection. In other words, we

assume that only a small proportion of features are contributing to classification task. This assumption is sometimes reasonable when we include a variety of features from all possible resources and there are some redundant features i.e. $d_j \approx 0$. Even if some features with $d_j \neq 0$ hold some information, it might be buried in a larger scale of noise. Hence, feature selection is useful and sometimes even necessary in many occasions.

3.2.1 FAIR

Features Annealed Independence Rule (FAIR) was proposed [13] for feature selection and classification in high dimensional setting. They measure the importance of features by t-statistics, which is defined as follow:

$$T_j = \frac{d_j}{\sqrt{S_{1j}^2/n_1 + S_{2j}^2/n_2}},\ j = 1, ..., p$$

where $S_{kj}^2 = \sum_{Y_i=k}(X_{ij} - \bar{X_{ij}})^2/(n_k - 1)$ is the sample variance of jth feature in class k. If we assume that the variance matrices of two groups are equal, than we will have a t-test with pooled variance:

$$T_j = \frac{d_j}{S_j^p},\ S_j^p = \sqrt{\frac{n_1 S_{1j}^2 + n_2 S_{2j}^2}{n_1 + n_2}}\ j = 1, ..., p$$

Either way the discriminant function of FAIR is:

$$\hat{\delta}_{FAIR} = \sum_{j=1}^{p} \frac{\hat{d_j}(x_j - \mu_j)}{\hat{\sigma}_j^2} \mathbb{1}(|T_j| > b)$$

where b is the threshold for feature selection. If the t_j exceed the threshold, we consider the corresponding jth feature significantly important and contributing to the classification and keep it in discriminant function, otherwise we delete this feature. Once important features are screened out, Features annealed independence rule is to apply independence classifier to the selected features.

Thresholding is also equivalent to selecting m_b features with largest $|t_j|$. Hence, we can also sort the features according to its t value first and the number of features can be determined by the following optimization problem[13]:

$$m_{opt} = arg \max_{1 \leq m \leq p} \frac{1}{\lambda_{max}^m} \frac{[\sum_{j=1}^m d_j^2/\sigma_j^2 + m(1/n_2 - 1/n_1)]^2}{nm/(n_1 n_2) + \sum_{j=1}^m d_j^2/\sigma_j^2}$$

where λ_{max}^m is the largest eigenvalue of the correlation matrix of the truncated features. The empirical estimate of this optimization with training data is:

$$\hat{m}_{opt} = arg \max_{1\leq m\leq p} \frac{1}{\hat{\lambda}_{max}^m} \frac{[\sum_{j=1}^m \hat{d}_j^2/s_j^2 + m(1/n_2 - 1/n_1)]^2}{nm/(n_1 n_2) + \sum_{j=1}^m \hat{d}_j^2/s_j^2}$$

Neither of these two ideas is time efficient. The threshold b can only be determined by cross-validation while the optimization problem to find the best m can only be solved by exhaustion, which is also time consuming in high dimensional space. However, these two ideas of features selection can combine to improve the computation efficiency. An available idea is to screen out all the important features with a reasonable threshold and find the best m_{opt} among these features.

3.2.2 NSC

[34] proposed the Nearest Shrunken Centroids (NSC) methods which can be used to identify a subset of features that best separate each class. Compared with FAIR, NCS is able to handle multi-class classification task. Denote the mean of jth feature for class k by $\bar{X}_j^k$ and the mean of jth feature for overall data by by $\bar{X}_j$. If we assume that all classes share same the covariance matrix, then the normalized deviation to overall centroid for each feature is

$$d_j^k = \frac{\bar{X}_j^k - \bar{X}_j}{\sqrt{\frac{1}{n_k} - \frac{1}{n}}(S_j + s_0)}$$

where s_0 is a positive constant and S_j is the pooled standard deviation for jth feature.

$$S_j^2 = \sum_{k=1}^K (n_k - 1)S_{kj}^2/(n-K) = \sum_{k=1}^K \sum_{Y_i=k} (X_{ij} - X_j^k)^2/(n-K)$$

The positive constant s_0 is included in case of sensibility caused by the denominator S_j being too small. The idea of shrunken centroid method is to shrink this deviation d_j^k by soft thresholding. More specifically, we have

$$\tilde{d}_j^k = s_\lambda^{soft}(d_j^k) = sign(d_j^k)(|d_j^k| - \lambda)_+$$

We can recover the centroids by shrunken deviation

$$\tilde{X}_j^k = \hat{X}_j + \sqrt{1/n_k - 1/n}(S_j + s_0)\tilde{d}_j^k$$

All deviations will be shrunken towards zero by at most λ and as a result, centroids of all groups will be closer to the overall centroid. Groups with deviations smaller than the threshold λ will be "deleted" because the centroids of these groups will be equal to the overall centroid and not contributing to the

classification function.

The shrinking process is also equivalent to the following form

$$\tilde{X}_j^k = s_{\triangle_j^k}(\bar{X}_j^k) = sign(\bar{X}_j^k)(|\bar{X}_j^k| - \triangle_j^k)_+$$

where $\triangle_j^k = \lambda\sqrt{\frac{1}{n_k} - \frac{1}{n}}(S_j + s_0)$, which is the denominator of $\tilde{d}_j^k$. If the data is normalized and all features have equal standard deviation 1 then $\triangle_j^k$ is same among features, $\triangle_j^k = \triangle^k$. Then we can simplify the shrinking process further.

$$\tilde{X}^k = s_{\triangle^k}(\bar{X}^k) = sign(\bar{X}^k)(|\bar{X}^k| - \triangle^k)_+$$

The threshold λ can only be selected by cross-validation.

3.2.3 SCRDA

Shrunken Centroids Regularized Discriminant Analysis (SCRDA)[18] is a more general approach compared with NSC. In both FAIR and NSC, we assume the independence of features and conduct either t-test or shrinking on features independently. However, the independence assumption is not valid most of time in practice. Hence, in SCRDA we remove the independence assumption and in order to obtain a non-singular and stable estimate of covariance matrix, we firstly regularize the empirical estimator of covariance matrix.

$$\tilde{\Sigma} = \alpha\hat{\Sigma} + (1-\alpha)I_p$$

Regularization can also be conducted on correlation matrix in a same way.

$$\tilde{R} = \alpha\hat{R} + (1-\alpha)I_p$$

where $\hat{R} = \hat{D}^{-1/2}\hat{\Sigma}\hat{D}^{-1/2}$, $\hat{D}$ is the diagonal matrix with diagonal elements of $\hat{\Sigma}$. we recover the covariance matrix estimate by $\tilde{\Sigma} = \hat{D}^{1/2}\tilde{R}\hat{D}^{1/2}$.

The second step is shrinking. Besides of shrink centroids on original features directly, another two projection were proposed in [18]: $\bar{X}^{*k} = \tilde{\Sigma}^{-1}\bar{X}^k$,or $\bar{X}_*^k = \tilde{\Sigma}^{-1/2}\bar{X}^k$. Then conduct shrinking process on projected scale and recover the shrunken centroids on original scale. By using projection based on covariance matrix, SCRDA took the correlation of features into consideration and if we assume the correlation matrix be identical matrix, SCRDA is equivalent to NSC. However, since the shrinkage is imposed on projected scale, it might not be as useful in features selection as NSC.

3.3 Projection

A similar idea to feature selection is projection. Rather than screen out important features directly, projection aims to find the best linear combinations of features to characterize and classify different classes.

3.3.1 PCA

Principle Component Analysis (PCA) aims to find the a set of linear combinations $\beta_j^T X$ of variables with maximum covariance. Since $var(\beta^T X) = \beta^T Cov(X)\beta$, assuming the covariance matrix Σ has eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$ and corresponding eigenvectors $e_1, \cdots, e_p$, we have optimization problem

$$\max_{\beta_j \in \mathscr{R}^p} \{\beta_j^T \Sigma \beta_j\} \quad \text{subject to } \beta_j^T \beta_j = 1$$
$$\beta_j^T \beta_l = 1,\ l = 1, \ldots, j-1$$

the solution to this problem is $\beta_j = e_j$ and the jth principle component is given by

$$Y_j = e_j^T X = e_{j1} X_1 + e_{j2} X_2 + \cdots + e_{jp} X_p$$

And these components have properties as follow

$$Var(Y_j) = e_j^T \Sigma e_j = \lambda_j \qquad j = 1, 2, \cdots, p$$
$$Cov(Y_j, Y_l) = e_j^T \Sigma e_l = 0 \qquad j \neq l$$

Among all linear combination of features $\{a^{'} x | a^{\prime} a = 1\}$, the first component has largest variance which is the largest eigenvalue λ_1. Moreover, the principle components gives a list of orthogonal linear combinations of features with descending variances. In other words, the principle components is the set of linearly uncorrelated variables from rotating or reorganizing original dimensions such that the data diverge decreasingly on each dimension of principle component. Hence the importance of contribution of components is also in descending order. Usually, we use the first two or three principle components to visualize high-dimensional data with maximum divergence. We can also use the data projected on the first several components in order to decrease dimensions.

Empirically, when the dimensions p is greater than the sample size n, the empirical estimator of covariance matrix is not full rank. In such cases, principle components can be useful to project original data to a lower dimension space where the features represented by principle components are uncorrelated and the covariance matrix is diagonal and not singular. LDA can also be implemented with principle components instead of using original variables to avoid singularity issue.

3.3.2 HDDA

High dimensional discriminant analysis (HDDA) [6] shares a similar idea to PCA and it can be considered as a modification of QDA using PCA. Besides of assuming that data from group i follow Multivariate normal distribution $N(\mu_k, \Sigma_k)$ as in QDA, it further imposes some assumptions on eigenvalues of Σ_k: except the first d_k eigenvalues, all the rest $p - d_k$ eigenvalues are same. In other words, assume Q_k is the orthogonal matrix of eigenvectors e_i^k of Σ_k, $\triangle_k = Q_k^T \Sigma_k Q_k$, which is a diagonal matrix containing the eigenvalues of Σ_k, should have the following form:

$$\triangle_k = \begin{pmatrix} \lambda_1^k & & 0 & & & \\ & \ddots & & & 0 & \\ 0 & & \lambda_{d_k}^k & & & \\ & & & b^k & & 0 \\ & 0 & & & \ddots & \\ & & & 0 & & b^k \end{pmatrix} = \begin{pmatrix} \Lambda_k & 0 \\ 0 & b^k I \end{pmatrix}$$

where $\lambda_i^k \geq b^k$ for $i = 1, \ldots, d_k < p$. In the view of principle components, HDDA assumes that the principle components can be split into two parts; the first part consisting of first d_k principle components with large variance while the second part includes the rest which can be considered with same small eigenvalues or variance b^k. In other words, $Q_k = (Q_{k1}, Q_{k2})$. Let $\tilde{Q}_k \in \mathscr{R}^{p \times p}$ be made of the first d_k columns of Q_k supplemented by zeros, i.e. $\tilde{Q}_k = (Q_{k1}, 0)$ and $\bar{Q}_k \in \mathscr{R}^{p \times p} = Q_k - \tilde{Q}_k = (0, Q_{k2})$, then we have

$$I = QQ^T = (\tilde{Q}_k + \bar{Q}_k)(\tilde{Q}_k + \bar{Q}_k)^T = \tilde{Q}_k \tilde{Q}_k^T + \bar{Q}_k \bar{Q}_k^T$$

HDDA defines a projection operator of x on the space spanned by first d_k components as follow:

$$P_k(x) = \tilde{Q}_k \tilde{Q}_k^T (x - \mu_k) + \mu_k$$

Now recall that in QDA the discriminant function is

$$\delta_{QDA}(x) = arg\max_{k \in \mathscr{Y}} \{2 \log \pi_k - \log |\Sigma_k| - ||x - \mu_k||^2_{\Sigma_k}\}$$

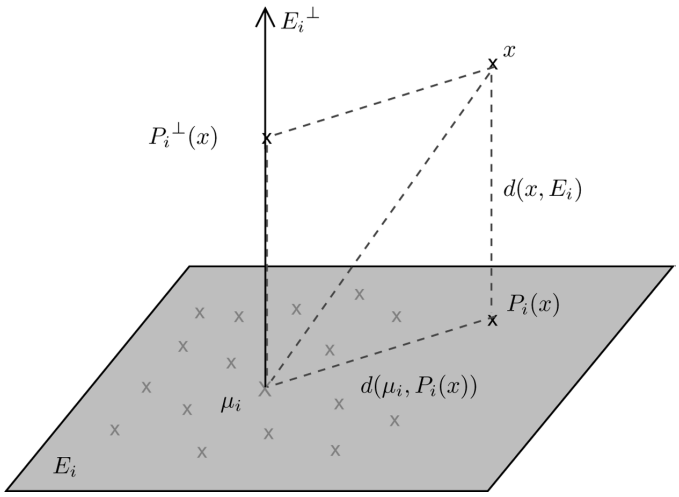

Figure 3.1: Distance decomposition figure from [6]

Then under the assumption in HDDA, the Mahalanobis distance term can be decomposed into two parts as follow:

$$\begin{aligned}
||x-\mu_k||^2_{\Sigma_k} &= (x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) \\
&= (x-\mu_k)^T Q_k \triangle_k^{-1} Q_k^T (x-\mu_k) \\
&= (x-\mu_k)^T (\tilde{Q}_k + \bar{Q}_k) \begin{pmatrix} \Lambda_k & 0 \\ 0 & b^k I \end{pmatrix}^{-1} (\tilde{Q}_k + \bar{Q}_k)(x-\mu_k) \\
&= (x-\mu_k)^T \tilde{Q}_k \Lambda_k^{-1} \tilde{Q}_k^T (x-\mu_k) + \frac{1}{b^k}(x-\mu_k)^T \bar{Q}_k \bar{Q}_k^T (x-\mu_k) \\
&= (x-\mu_k)^T \tilde{Q}_k \Lambda_k^{-1} \tilde{Q}_k^T (x-\mu_k) + \frac{1}{b^k}(x-\mu_k)^T (I - \tilde{Q}_k \tilde{Q}_k^T)(x-\mu_k) \\
&= ||\mu_k - P_k(x)||^2_{\mathscr{A}_k} + \frac{1}{b^k}||x-P_k(x)||^2
\end{aligned}$$

where $\mathscr{A}_k = \tilde{Q}_k \triangle_k^{-1} \tilde{Q}_k^T$. Then the discriminant function will be

$$\begin{aligned}
\delta_{HDDA}(x) = arg\max_{k\in\mathscr{Y}} \{ & 2\log\pi_k - \sum_{i=1}^{d_k} \log\lambda_i^k - (p-d_k)\log b^k \\
& - ||\mu_k - P_k(x)||^2_{\mathscr{A}_k} - \frac{1}{b^k}||x-P_k(x)||^2 \}
\end{aligned}$$

The decomposition can be interpreted as shown in Figure 3.1.

In the paper[6], there is also some flexibility on the assumption such as common principle dimensions i.e. $d_k = d$, common eigenvalues i.e. $\lambda_i^k = \lambda, b^k = b$, common eigenvectors i.e. $Q_k = Q$ or common covariance matrix i.e. $\Sigma_k = \Sigma$ among groups. Hence, there are in total 28 variant model listed in the paper.

For different models under different assumptions, the estimator of parameters might be slightly different. However, the key parameter in HDDA model is the intrinsic dimension d_k. If d_k are common among classes, we can determine the dimension by cross-validation. Otherwise we use "scree-test" approach by [9] based on the eigenvalues of within-class covariance matrix Σ_k. Briefly speaking, we compare the difference between two subsequent eigenvalues with a threshold t and determine the dimension when the difference is smaller than threshold. It still needs cross-validation on the training set to determine this threshold λ.

In summary, this method is a combination of QDA and PCA. However, instead of using the first several components directly, HDDA tried to take advantage of the remaining components by assuming equal variance on them. Hence, the true variances of these remaining components, which are empirically zero as a result of singularity of empirical covariance estimator, will be recovered and no bias is imposed on the first several components.

3.3.3 Fisher

In either PCA or HDDA, we take advantage of within-class covariance matrix and find the projection based on the eigen-vectors. Different from previous approaches, Fisher method aims to find projection that distinguishes classes best by using between-class covariance matrix, which is the covariance matrix of the group centroids, defined as follow.

$$\Sigma_B = Cov_{Y \in \mathscr{Y}}(\mu_Y) = \sum_{k=1}^{K} (\mu_k - \mu)(\mu_k - \mu)^T \pi_k$$

Fisher's method seeks a low-dimension projection of the observations such that the between-class variance is large relative to the within-class variance [11], which is the solution to the following problem

$$\max_{\beta \in \mathscr{R}^p} \{\beta^T \Sigma_B \beta, \text{ subject to } \beta^T \Sigma_W \beta \leq 1\} \tag{3.1}$$

If we substitute $\tilde{\beta} = \Sigma_W^{1/2} \beta$, where $\Sigma_W^{1/2}$ is the symmetric square root of Σ_W, then the problem above is equivalent to the following one

$$\max_{\tilde{\beta} \in \mathscr{R}^p} \{\tilde{\beta}^T \Sigma_W^{-1/2} \Sigma_B \Sigma_W^{-1/2} \tilde{\beta}, \text{ subject to } \tilde{\beta}^T \tilde{\beta} \leq 1\} \tag{3.2}$$

Similar to seeking principle components, we can find a set of orthogonal vectors β_k that maximize the objective function decreasingly and they are called fisher discriminant vectors.

$$\begin{aligned} &\max_{\beta_k \in \mathscr{R}^p} \{\beta_k^T \hat{\Sigma}_B \beta_k\} \\ &\quad \text{subject to } \beta_k^T \hat{\Sigma}_W \beta_k \leq 1 \\ &\qquad \beta_k^T \hat{\Sigma}_W \beta_l = 0,\ l = 1, \ldots, k-1 \end{aligned} \tag{3.3}$$

Note that the rank of between-class covariance matrix is no greater than $K-1$. Hence, we can only get no more than $K-1$ fisher discriminant vectors.

Empirically, the between-class covariance matrix can be estimated by $\hat{\Sigma}_B = \frac{1}{n}\sum_{k=1}^{K} n_k \hat{\mu}_k \hat{\mu}_k^T$ (if the data are scaled to the center in advance). However, for within-class covariance matrix, same issue happens in high dimensional space where sample size n is usually less than dimensions p as mentioned earlier. From 3.1 to 3.2 we require the inverse of Σ_W while the empirical estimator is singular. Actually, vectors in the null space of $\hat{\Sigma}_W$ but not in the null space $\hat{\Sigma}_B$ can make the objective function positive infinity.

In order to address the issues caused by singularity of empirical estimator $\hat{\Sigma}_W$, some modifications have been introduced. [27] imposed another constrain $\beta^T \beta = 1$ on the optimization problem 3.1; [33] required that β should not be in the null space of $\hat{\Sigma}_W$, i.e. $\beta^T \hat{\Sigma}_W \beta > 0$; [16], [12] assumed the independence features with diagonal within-class covariance matrix, which can be supported by the conclusion from [1] who compared the traditional LDA and Naive Bayes classifier.

3.3.4 SDA

Sparse Discriminant Analysis proposed by is a modification on fisher method [11]. It aims to find sparse discriminant vectors by solving the optimization problem as follow

$$\begin{aligned} &\max_{\beta_k \in \mathscr{R}^p} \{\beta_k^T \hat{\Sigma}_B \beta_k - \gamma \|\beta_k\|_1\} \\ &\quad \text{subject to } \beta_k^T \tilde{\Sigma}_W \beta_k \leq 1 \\ &\qquad \beta_k^T \tilde{\Sigma}_W \beta_l = 0,\ l = 1, \ldots, k-1 \end{aligned}$$

where $\tilde{\Sigma}_W = \hat{\Sigma}_W + \Omega$ with Ω a positive definite matrix and λ is the tuning parameter for penalty. The biased estimator $\tilde{\Sigma}_W$ was used in [19] to avoid the singularity of empirical estimator.

However, this is not a convex problem. In order to solve this one, they instead apply L_1 penalty to the optimal scoring formula for Fisher's method. The original optimal scoring criterion takes the form as follow

$$\begin{aligned} &\max_{\beta_k \in \mathscr{R}^p, \theta_k \in \mathscr{R}^K} \{\|Y\theta_k - X\beta_k\|^2\} \\ &\quad \text{subject to } \frac{1}{n}\theta_k^T Y^T Y \theta_k = 1 \\ &\qquad \theta_k^T Y^T Y \theta_l = 0,\ l = 1, \ldots, k-1 \end{aligned} \tag{3.4}$$

where Y denotes an $n \times K$ matrix of dummy variables for K classes i.e. Y_{ik} is an indicator for whether the ith observation is labeled as class k, θ_k is a scores vector of length K and β_k is consistent with that in Fisher's method. Then the Sparse discriminant analysis is to impose l_1 penalty on 3.4

$$\max_{\beta_k \in \mathscr{R}^p, \theta_k \in \mathscr{R}^K} \{\|Y\theta_k - X\beta_k\|^2 + \gamma\beta_k^T \Omega \beta_k + \lambda \|\beta_k\|_1\}$$
$$\text{subject to } \frac{1}{n}\theta_k^T Y^T Y \theta_k = 1$$
$$\theta_k^T Y^T Y \theta_l = 0,\ l = 1, \ldots, k-1$$

3.3.5 PLDA

The penalized LDA proposed by [38] is another modification on fisher's method by imposing penalty functions on discriminant vectors with a more general and flexible form.

$$\max_{\beta_k \in \mathscr{R}^p} \{\beta_k^T \hat{\Sigma}_B \beta_k - P_k(\beta_k)\}$$
$$\text{subject to } \beta_k^T \tilde{\Sigma}_W \beta_k \leq 1$$
$$\beta_k^T \tilde{\Sigma}_W \beta_l = 0,\ l = 1, \ldots, k-1$$

where $\tilde{\Sigma}_W$ is a positive definite estimator for Σ_W and $P(\beta)$ is a convex penalty function. If we substitute $P_k(\beta_k)$ with $\gamma\|\beta_k\|_1$ then the objective function is exactly same as SDA. However, different from SDA, the penalty used in PLDA [38] can be either L_1 type

$$P_k(\beta_k) = \lambda_k \sum_{j=1}^{p} |\hat{\sigma}_j \beta_{kj}|$$

or fused lasso proposed by [35]

$$P_k(\beta_k) = \lambda_k \sum_{j=1}^{p} |\hat{\sigma}_j \beta_{kj}| + \gamma_k \sum_{i=2}^{p} |\hat{\sigma}_j \beta_{kj} - \hat{\sigma}_{j-1} \beta_{k,j-1}|$$

where λ and γ are tuning parameter for penalty and can be chosen by cross-validation.

The penalty in PLDA is consistent with that in SDA if the data is normalized and $\hat{\sigma}_j = 1$. The fused lasso penalty is designed to take advantage of sparsity in both coefficients β_{kj} and $\hat{\sigma}_j \beta_{kj} - \hat{\sigma}_{j-1} \beta_{k,j-1}$, the difference between coefficients for two consecutive $j-1$th and jth features in one projection dimension β_k, .i.e. "flatness of the coefficient profiles β_{kj} as a function of j"[35], with the scale of these two features taken into consideration. Hence, this penalty is only used when the features are in a meaningful order with consistency between sequential features.

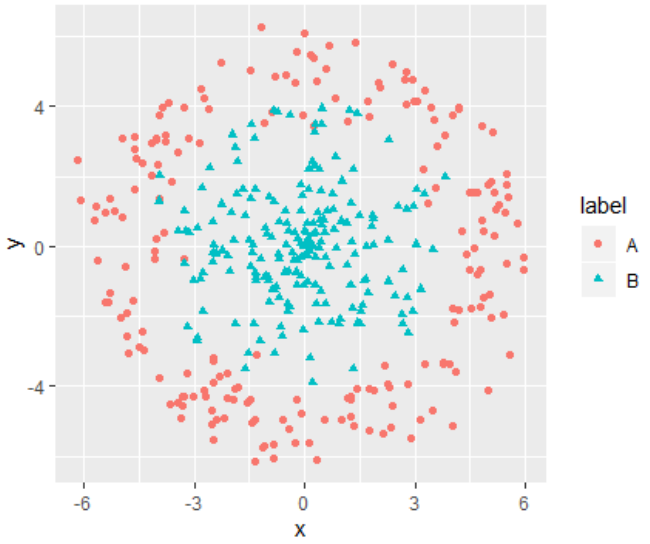

Figure 3.2: Linearly inseparable example

3.4 Nonlinear classification

There are still cases when groups are not linearly separable. A classical example is shown as Figure 3.2. In this case, the centroids of these two groups overlap while the radius is different. Apparently these two groups are separable but no linear classifier would work by using raw features. In this section, we introduce several well known nonlinear classification models and how the non-linearity is achieved.

3.4.1 SVM and Kernels

Support Vector Machine proposed by [5] has achieved massive success and still keeps popular in many real-world situations especially in high-dimensional classification tasks. The main idea of SVM is to find the best hyperplane that separates two groups of data. So basically it is still a linear classifier. Assuming $x_i \in \mathscr{R}^p$ and $y_i \in \{-1,1\}$, a hyperplane was defined as $\{x: f(x) = x^T\beta + \beta_0 = 0\}$ where β is a unit vector i.e. $\|\beta\| = 1$. Then the classification function is $g(x) = sign(x^T\beta + \beta_0)$.

In order to find the "best" hyperplane, we introduce the definition of margin. The functional margin of an observation (x_i, y_i) with respect to a hyperplane $f(x) = 0$ is defined as

$$M_i = y_i f(x_i) = y_i(x_i^T\beta + \beta_0)$$

Recall that the geometric distance from a point xi to the hyperplane $x^T\beta + \beta_0 = 0$ is $|x_i^T\beta + \beta_0|/\|\beta\|$, M_i measures the distance from an observation to the classification hyperplane when $\|\beta\| = 1$. Hence, we aim to find the hyperplane maximizing the minimum margin, which is equivalent to the optimization problem

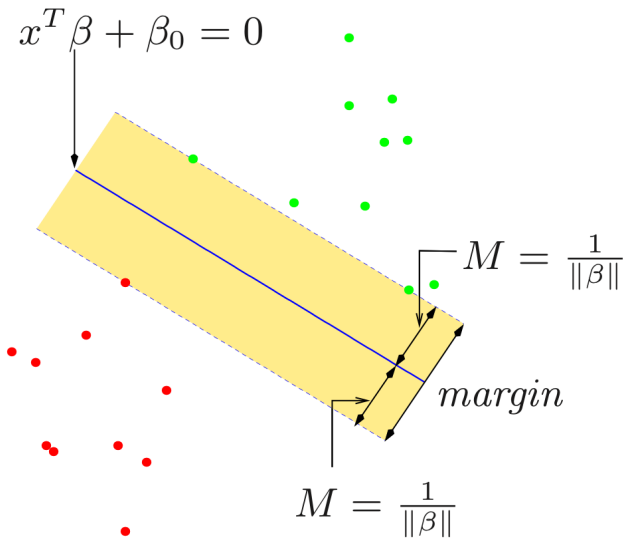

Figure 3.3: SVM illustration from [20]

below

$$\max_{\beta \in \mathscr{R}^p, \beta_0 \in \mathscr{R}, \|\beta\|=1} M$$
$$\text{subject to } M_i \geq M,\ i = 1, ..., n$$

Margins of observations from either positive or negative group should be no less than M, so then sum of two shortest margins from two groups $2M$ is defined as the margin of the training set with respect to the hyperplane[29].

Alternatively, we can drop the norm constraint on β and fix $M = 1$, then the optimization problem is equivalent to the following form

$$\min_{\beta \in \mathscr{R}^p, \beta_0 \in \mathscr{R}} \|\beta\|$$
$$\text{subject to } M_i \geq 1,\ i = 1, ..., n$$

When two clusters are not linearly separable or even overlap, a soft constrain is introduced to allow for some points on the wrong side of the hyperplane. As shown below, the optimization problem is modified

by introducing the slack variables ξ.

$$\begin{aligned} \min_{\beta \in \mathscr{R}^p, \beta_0 \in \mathscr{R}} & \|\beta\| \\ \text{subject to } & M_i \geq 1 - \xi_i, \ i = 1, ..., n \\ & \xi_i \geq 0, \ i = 1, ..., n; \\ & \sum_{i=1}^{n} \xi_i \leq F \end{aligned}$$

where F is a constant controlling the total amount of slackness we allow. By introducing slack variables ξ, we allow small margins close to the hyperplane with $\xi \in [0,1]$ and even negative margins i.e. $yf(x) < 0$ with $\xi \geq 1$. In order to solve the optimization problem, the objective function $\|\beta\|$ is always replaced with the equivalent one $\frac{1}{2}\|\beta\|^2$ for convenience. Obviously, this is a convex problem and the only local minimum should be the optimal solution. To find the solution, we firstly get the Lagrangian (Primal) as follow

$$L((\beta_0, \beta, \xi), C, \alpha, \mu) = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{n} \alpha_i (M_i - (1 - \xi_i)) - \sum_{i=1}^{n} \mu_i \xi_i + C \sum_{i=1}^{n} \xi_i \tag{3.5}$$

where α_i, μ_i, C are Lagrange multiplier for the constrains and ξ_i are feasible variables and all of them must be positive. Since the constant is independent of any variables, we usually drop the constant term. By setting the derivatives w.r.t. β_0, β, ξ to zero we have [20]

$$\begin{aligned} 0 &= \sum_{i=1}^{n} \alpha_i y_i, \\ \beta &= \sum_{i=1}^{n} \alpha_i y_i x_i, \\ \alpha_i &= C - \mu_i, \end{aligned}$$

The equations above along with the following constrains:

$$\begin{aligned} \alpha_i (M_i - (1 - \xi_i)) &= 0 \\ M_i - (1 - \xi_i) &\geq 0 \\ \mu_i \xi_i &= 0 \\ \xi_i &\geq 0 \\ C(\sum_{i=1}^{n} \xi_i - F) &= 0 \\ \sum_{i=1}^{n} \xi_i - F &\leq 0 \end{aligned}$$

are called Karuash-Kuhn-Tucker conditions. By the Karuash-Kuhn-Tucker conditions above, we can cancel β, β_0 in M_i, C, μ_i and ξ_i and obtain the Lagrangian dual as below

$$L = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

By maximizing the dual problem subject to Karuash-Kuhn-Tucker conditions, we can find the unique solution α_i^* and the hyperplane $\{x : f(x) = \sum_{i=1}^{n} \alpha_i^* y_i x_i^T x + \beta_0^* = 0\}$. The hyperplane is only determined by a linear combinations of $y_i x_i$ and those observations with non-zero weights $\alpha_i \neq 0$ is called support vectors.

So far the optimization is still searching for a linear classifier with the hyperplane defined as $\{x : f(x) = x^T \beta + \beta_0 = 0\}$ a linear from with respect to x. The "Kernel trick" makes it possible for linear boundaries to lie in a higher dimension space and to be nonlinear projected on original space[20]. We define the basis functions as $h_m(x) =, m = 1, ..., M$ and represent all observations in the new space $h(x_i) = (h_1(x_i), h_2(x_i), ...h_M(x_i))$. The hyperplane in this new space is $\{x : f'(x) = f(h(x)) = \sum_{i=1}^{n} \alpha_i^* y_i h(x_i)^T h(x) + \beta_0^* = 0\}$ and the classification function is still $g(x) = sign(f'(x))$. As observed from the form of hyperplane, the equation is a linear combination of the inner product of $h(x_i)$ and $h(x)$, which is called kernel function.

$$\mathscr{K}(x, y) = h(x)^T h(y) \tag{3.6}$$

There are many types of kernel functions such as polynomial, Gaussian, radial basis[20] and splines[21]. Since we basically using linear classifier in simulation and experiment, we will not go further with the introductions on kernels. But the idea of using kernel trick or in other words basis expansion can be drawn into our conventional linear classification model.

SVM is also capable of multi-classification and there are three ways based on binary classifiers: "one-against-all", "one-against-one", and Directed Acyclic Graph Support Vector Machines (DAGSVM) [22]. The first one builds K SVM classifiers for K classes respectively with data from k-th class as positive and from other classes as negative. Then we take the margins as score and we classify a new observation with the largest margin. The second method constructs $K(K-1)/2$ classifiers where each one is trained on data from two classes pair-wisely. For a new observation, we use the voting strategy: with votes for all classes initialized as zero, we add one to k-th class's vote if the result from one of these classifiers is k-th class, then the conclusion is the class with greatest votes. Though shares the same training steps as "one-against-one" method with $K(K-1)/2$ SVM classifiers, the third one DAGSVM is more complicated and delicate with a tree based decision function.

One thing we need to pay attention to is its sensitivity to prior. As mentioned in [14], SVM is sensitive to both noise and prior. It tends to label the new observation into the class with more samples. This is somehow undesirable when we care more about the minority instead of majority. This property is

illustrated in simulation where we checked the sensitivity of all methods we used.

3.4.2 kNN

kNN is one of the most basic non-linear classifiers and the idea of kNN is relatively simple. Given the training data, for a new sample, we find the k samples from the training data set that are closest to it. These samples are called k nearest neighbors [14]. We predict the output value by taking the average of labels from these nearest neighbors if the label is a numeric variable

$$\delta(X) = \frac{1}{k} \sum_{X_i \in N_k(X)} Y_i$$

or taking the most voted label if it is categorical.

$$\delta(X) = \max_j \sum_{X_i \in N_k(X)} \mathbf{1}(Y_i = j)$$

This means kNN can be used in both regression and classification. However, in high dimensional space, kNN is not a good choice. This is also the origin of term "curse of dimensionality"[10]. As a non-parametric model, kNN requires large size sample to "full-fill" its input space the the required sample size grows exponentially with the increase of dimension.

3.4.3 Neural network

Similar to kNN, Neural network also requires a massive amount of samples. Neural methods have wide successful applications in high dimensional space and a wide range of variants in different scenarios such as CNN in image processing and RNN in for sequential data such as texts or time series. The complex structure and nonlinear activation function of Neural methods make is possible to do non-linear classification. At the same time, due to the complexity and huge amount of parameters in the model, we usually require huge data set to train the model and avoid over-fitting. In this thesis, we focus on situations when the sample size is relatively small. Hence, we won't discuss the neural methods in details.

Chapter 4

Numerical Results

In this section, we presented result for simulated and real data sets to illustrate the performance of our approach. The methods we compare with includes LDA [37], Naive Bayes, HDDA, (SC)RDA, PLDA and SVM. The measurement we use to evaluate the performance of classifier is accuracy.

4.1 Simulation

In this experiment, we assumed there are two groups $X_1, X_2 \in \mathscr{R}^p,\ p = 100$, following multivariate normal distribution with same covariance as shown in figure 4.1, i.e. $X_1 \sim N(0,\Sigma),\ X_2 \sim N(t,\Sigma)$, where t is a p dimensional vector with all elements equal and all diagonal entries in Σ are 1. Hence, all features are expected to contribute equally. In practice, t range from 0 to 3 since we found that all methods would reach high accuracy after 3. The covariance matrices we used in data generation cover four types of sparse covariance matrix: Tri-Diagonal, Auto-Regressive, Banded and Random Sparse Matrix as introduced in section 2.2.

In order to simulate the situation where the sample size is smaller then the dimension p, we only generate 10 samples from each group with 9 of them (balanced case) composing training set and 1 sample as test set. We repeated this procedure for multiple times to alleviate the variation caused by randomness of small sample size.

4.1.1 False positive rate and thresholding operator

We first check the differences among LDAS using different levels of false positive rate. In theory, the smaller false positive rate will lead to a covariance estimator closer to a diagonal one since the threshold is larger. The numerical result is consistent with our expectation. The middle plot in Figure 4.2 shows the accuracy lines for different levels of false positive rate $\rho = 0.1, 0.05, 0.01, 0.001$. Since the tri-diagonal matrix is really close to the diagonal matrix with large sparsity, the classifier with smallest false positive

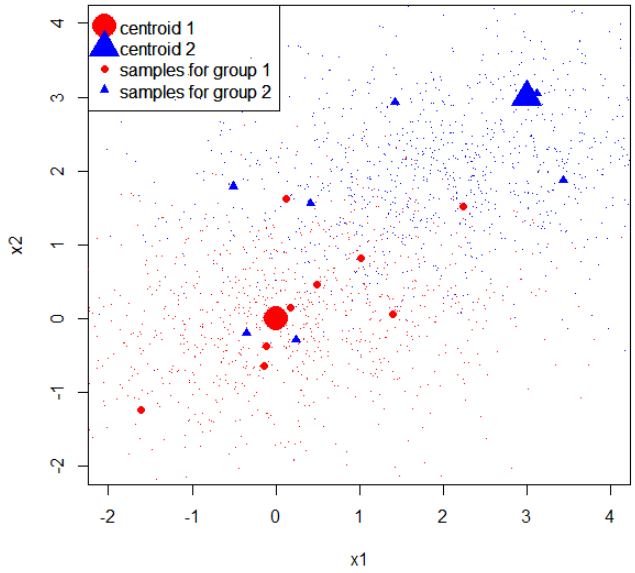

Figure 4.1: This figure illustrates data generation in first two dimensions. In each dimension the distance between two centroids is 3.

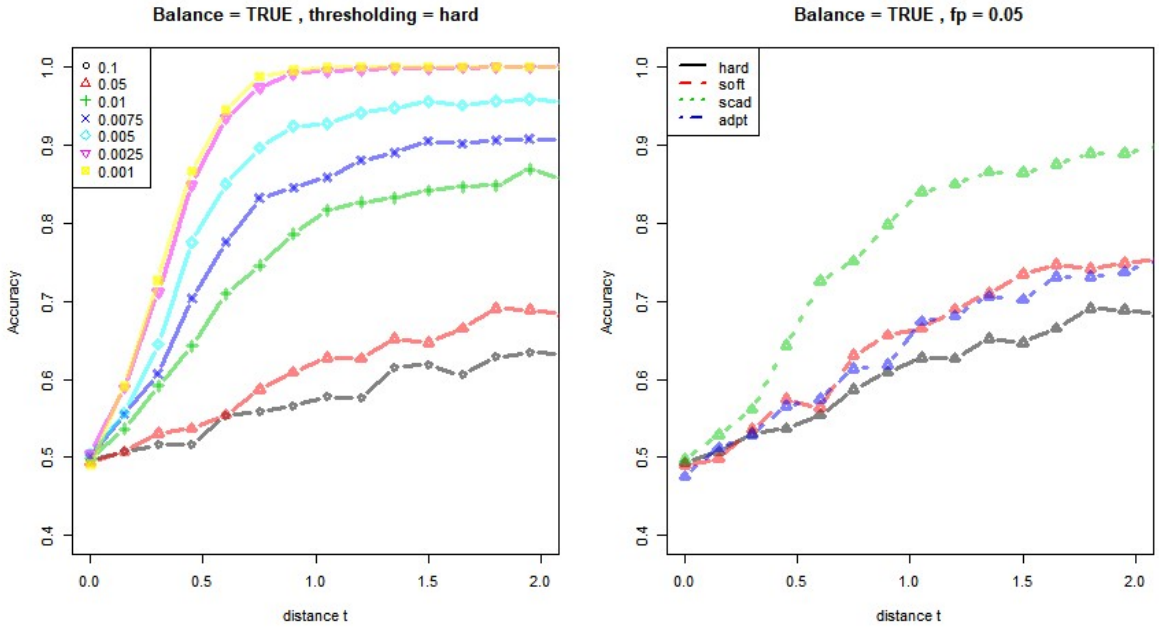

Figure 4.2: The left graph shows the accuracy of LDAS using different false positive rates with hard thresholding operator, and the right one shows accuracy using four types of thresholding operators with false positive rate $\rho = 0.05$, given data generated with tri-diagonal covariance matrix. The horizontal axis is the distance between two centroids of two groups (ranging from 0 to 2) and the vertical axis is the accuracy of classification.

rate achieved highest accuracy over the range of distance and the accuracy dropped with increase of the false positive rate.

The plot on the right hand illustrates how type of thresholding operator influences the performance of LDAS. Given a fixed false positive rate at 0.05, SCAD thresholding operator leaded to the highest accuracy while the hard one resulted in the worst. This is consistent with Figure 2.3 where the SCAD introduced little bias for large input and more shrinkage for small values. SCAD or Adaptive LASSO being the best operator can be considered as a sign of small threshold or, in other words, large positive rate and we should try smaller false positive rate with hard thresholding operator. For instance, if we compare the two green lines in the two graphs on the right side, we can observe an approximate level of accuracy over the whole range, which indicates that using hard thresholding with false positive rate $\rho = 0.01$ is approximately equivalent to using SCAD with $\rho = 0.05$. Figure 4.3 might give us some inspiration about the reason behind by illustrating the relationship between hard and SCAD thresholding. As we can see, the hard thresholding function with large threshold is quite close to the SCAD function with small threshold. If we continue to look at the performance with smaller false positive rate (Figure 4.6), we can find that as the false positive rate decrease, the best performing operator switch from adaptive LASSO to Hard thresholding and when Hard thresholding becomes the best one the false positive rate result in the highest accuracy.We also obtained the same pattern in simulations using other type of sparse covariance matrix in Figure 2.1. Hence, the rank of accuracy among different operators can be used as a guide for parameter tuning: we start from a relatively large false positive rate and shrink it until the Hard thresholding operator achieves highest overall accuracy.

4.1.2 Comparison with other methods

We compared our approach with other previous methods such as LDA [37], Naive Bayes, HDDA, RDA, PLDA and SVM. Since the simulated data follow a multivariate normal distribution and the expected classification hyperplane is expected to be linear, we didn't use any non-linear kernels for SVM to avoid over-fitting. Figure 4.5 shows how accuracy of each baselines change with increase of centroids distance. The curves of several strong baselines: PLDA, SVM and Naive Bayes overlap with our approach while other methods such as RDA and HDDA achieved lower accuracy overall. We also repeated the simulation with different type of sparse covariance matrix. The result is listed in Appendix.

4.1.3 Sensitivity to prior

The sensitivity of prior refers to that prior always induces a sensitive model to classify the sample as the group of large size with greater possibility than it should be. It happens when the data set is unbalanced. If we have control on experiment design, we might take it into consideration and try to avoid unbalanced cases. However, there are more cases when the experiment is done and we only have access to the final

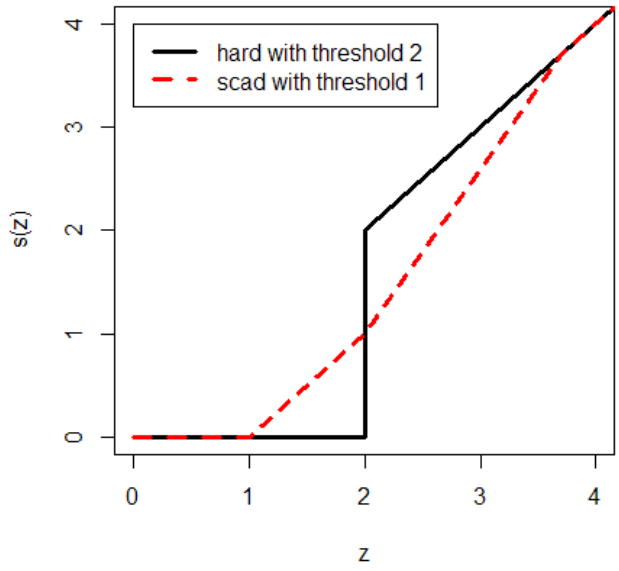

Figure 4.3: Relationship between hard and SCAD thresholding

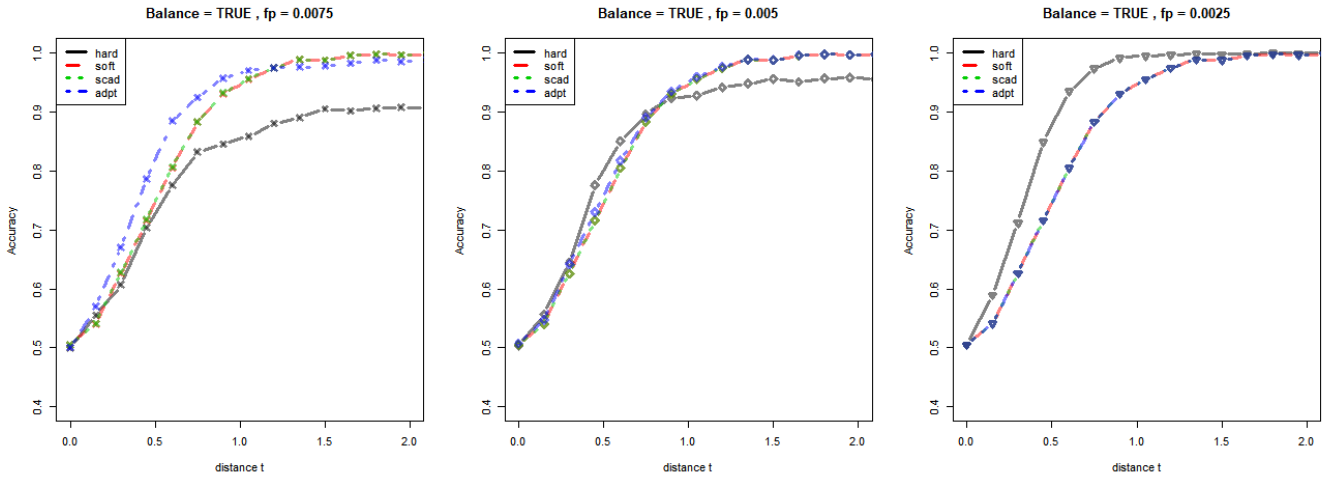

Figure 4.4: Accuracy curves of 4 type of operators given false positive rate at decreasing level.

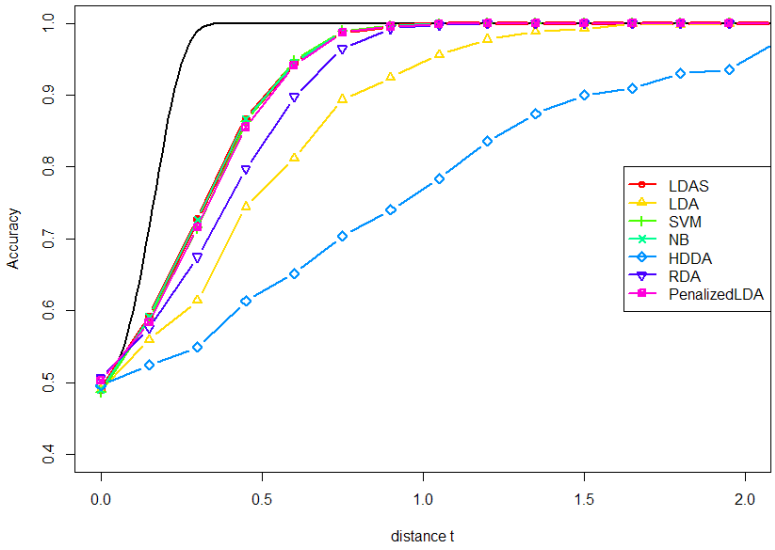

Figure 4.5: Accuracy comparison among different methods. The black line indicates the Bayes error

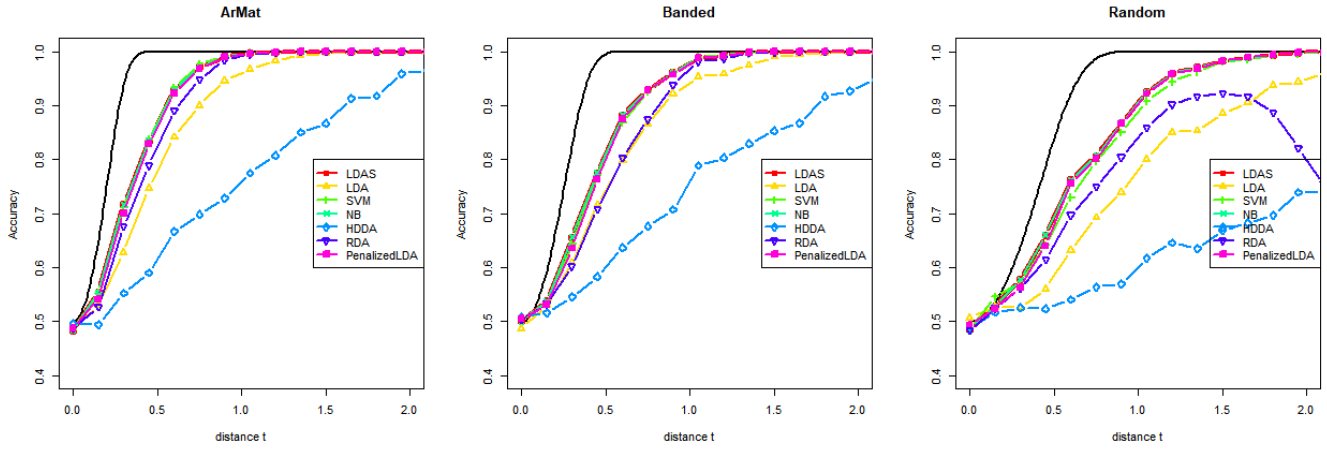

Figure 4.6: Accuracy curves comparison among methods with data generated from three type of sparse covariance matrices

result. The reason why we care about this sensitivity origins in two possible cases. In one possible case, we are interested in every group equally while we have to pour the unbalanced data set into our model in order to take full advantage of information. This is necessary especially in high dimensional where the sample size is always limited compared with dimensionality. Another case is when we have a large control group but a limited size of treatment group due to factors such as financial cost or implementing difficulty. However, we are more interested in the smaller treatment group and it is crucial to use the model insensitive to the prior.

Simulations so far are balanced cases where the training sample size for either class is 9. In order to test the sensitivity to prior, we adjusted the simulation by setting 10 samples for class 1 and 9 samples for class 2 in training set and the test set is an observation from class 2, which is the smaller group. The result is shown in Figure 4.7. The most sensitive baseline is SVM, whose average accuracy is even less than 20%. However, as the distance of two centroids increases, the accuracy of SVM gets improved significantly and catch up other baselines at distance around 1. Similar issue happens when using RDA and PLDA as well, though less serious than SVM. Both LDA and Naive Bayes are insensitive to prior and when two groups can not be distinguished from each other i.e. zero distance between two centroids, the accuracy is around 0.5 which is equivalent to random guessing with little influence from prior. Our approach is designed as the balance between these two methods and insensitive to prior as expected as well.

4.2 Real data

In this section we used two cancer data set which includes thousand of gene expression features. These two data sets are similar with respect to sample size, dimensionality, number of categories and balance. However, we observed a considerably different result which will be discussed in the following subsections.

4.2.1 Small Round Blue-Cell Tumor Data

We first considered the data set from the small round blue-cell tumor (SRBCT) micro-array experiment [25]. Accurate diagnosis of SRBCTs is the key to decisions on treatment options, responses to therapy and prognoses. However, there are four type of tumors denoted as EWS, BL-NHL, NB RMS and these tumors are hard to distinguish by light microscopy either manually or automatically. Alternative approaches were used for this situation and gene expression is one of them. The difficulty of gene analysis lies in the ultra high dimensionality of gene features. For instance, in the data used in [25] there are 83 SRBCTs samples of gene expression labeled with one of the four types and also 5 non-SRBCT samples, while each sample of either group consists of 6567 genes and even after filtering there are still 2308 genes, which is definitely a massive amount compared with the sample size. The diagnostic model in [25] consists of PCA for dimension reduction at the first stage and Artificial Neural Network[4] (ANN) for classification.

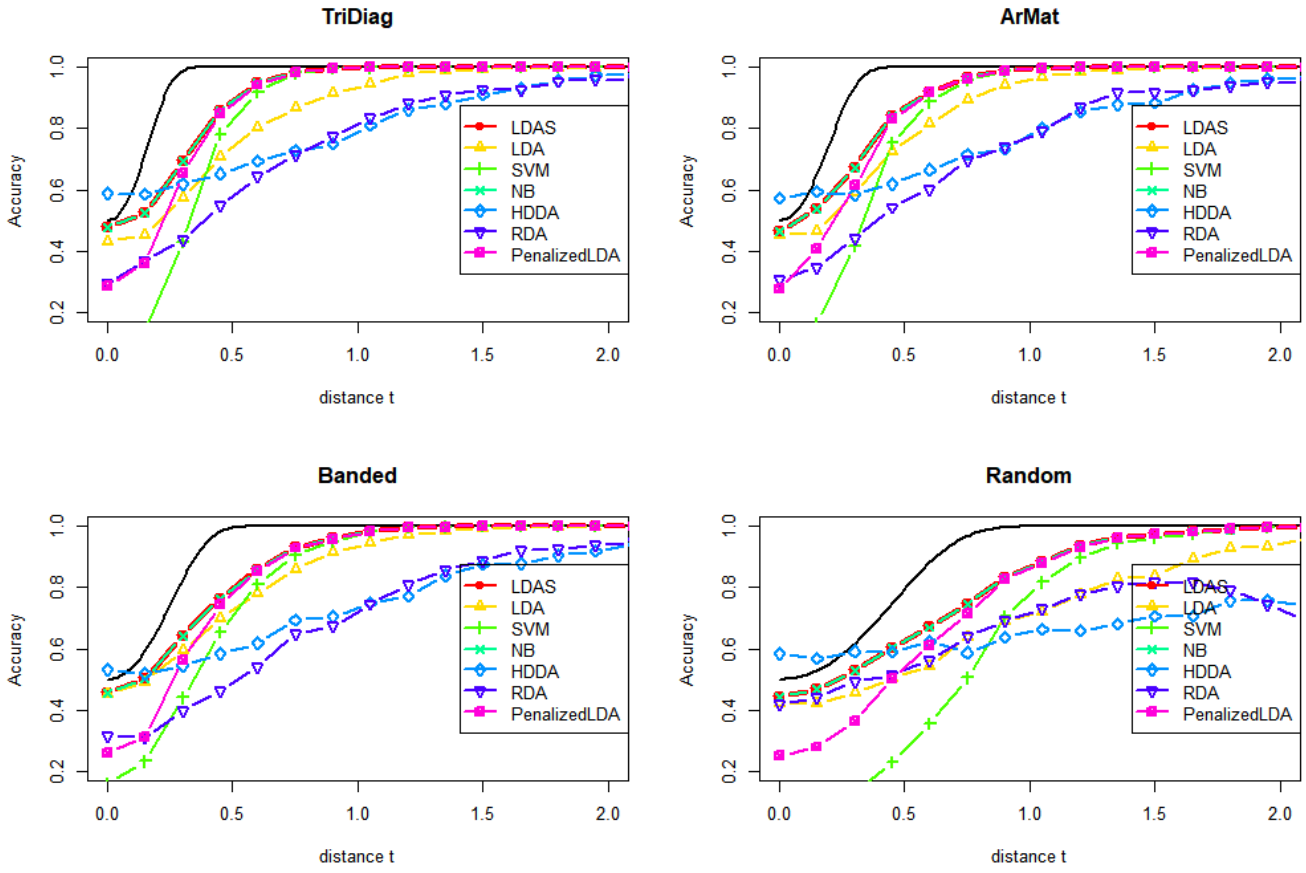

Figure 4.7: Accuracy curves comparison among different methods in unbalanced cases.

The final conclusion seems perfect with all samples correctly classified. However, the model was trained with data including only four types of SRBCT samples and as a result non-SRBCT patient can not be classified directly. They also set a 95 percent voting criteria for distinguish SRBCT and non-SRBCT samples but after implementing this thresholding rule some SRBCT samples was able to detected and the perfect accuracy was cracked.

Hence, in our experiment, we directly merged the train and test set as a whole which included all five groups (4 SRBCT groups and 1 non-SRBCT group). We split the whole data set into 10 folds and implemented 10-folds cross validation and kept all the ten accuracy of ten validation set to diminish the randomness caused by the train-test splitting. We tried four types of thresholding operators and four level of false positive rate $\rho = 0.1,\ 0.05,\ 0.001,\ 0.001$ as we did in simulation to verify the conclusion we got. Same baselines were tested with same cross validation setting.

The box-plots of accuracy are shown in Figure 4.8. For each box-plot, we implemented the experiment by randomly selecting part of the feature set (all models shared the same set of features) so that we can check how dimensionality influence the performance of methods. First, if we look at the box-plots on the top where only 100 features were selected, we can find some methods already achieved high accuracy, such as our LDAS method using SCAD thresholding operator and large false positive rate $\rho = 0.1, 0.05$, adapt operator and smaller false positive rate $\rho = 0.015$ and hard operator with smallest false positive rate. This is consistent with what we found in simulation. The hard one is expected to be the best operator that imposes no values greater then threshold. When either SCAD and adaptive lasso operator outperforms the hard one, we should turn down the false positive rate to improve the performance of hard one. This phenomenon is also clear in other three box-plots, where the accuracy of hard operator increased with smaller false positive rate while SCAD and adaptive ones often dropped at a small false positive rate. There are also cases when the highest accuracy at smallest false positive rate was from LDAS using adaptive operator (in the third and forth one) instead of hard one. This is reasonable since with the increase of dimensionality of input features, we expect more sparsity in covariance matrix, hence a false positive rate even less than 0.001. In summary, we manage to verify the previous conclusion that we should use smaller false positive rate until LDAS using hard operator has the best performance. Also, from this experiment we can also conclude that the higher dimensionality is, the smaller the best false positive rate should be, i.e. the more sparsity we should assume in covariance matrix.

If we compare our method with the baselines, we can see that using sparse estimator did improve the traditional LDA, which achieved the lowest accuracy overall. Similar to the result in simulation, Naive Bayes still have the highest accuracy among baseline methods and PLDA is still competitive to Naive Bayes. However, in this case SVM turned out not as effective as Naive Bayes. This can be explained by unbalance.

Compared with the ANN approach in [25], either our approach or Naive Bayes reached competitive performance with high accuracy and even 100% accuracy in some train-validation splittings. The variation

of the accuracy in the box-plots resulted from the randomness of train-validation allocation. Hence, even though ANN was reported to classified all samples correctly, it could be a random but perfect result and it's not convincing to claim a general conclusion.

4.2.2 Lung Cancer Dataset

The second data set we used is also related to cancer and gene expressions [32]. There are in total 65 samples covering three type of lung cancers:AD (adenocarcinoma), SQ (squamous), CAD(cell lines adenocarcinoma) and 17 healthy samples labeled NL (normal lung), each of which consists of 9036 gene features. This data set is similar to the SRBCT data set with respect to sample size and dimensionality of input space. We did the same 10 folds cross-validation testing with different range of features selected among the whole dimensions as we did for SRBCT data set; however, We observed a different pattern of accuracy among different methods. The results are shown in Figure 4.9.

The most obvious and surprising result is that the traditional LDA worked perfectly in all cases whatever the the dimensionality is and better than almost all other methods. This is totally different from the result of either simulation or SRBCT data set. In theory, we expect sparse correlation among features while simple LDA might "misunderstand" the correlations with a singular covariance estimator with a limited size of sample. One possible reason behind this surprisingly good result can be that there is a massive collinearity among these genes features, i.e. the input features lies in a lower dimensional space. As discussed in section 2.1.5, "lda" model in "MASS" classifies samples in the subspace with projection basis being eigenvectors of positive eigenvalues of the empirical estimator. The empirical estimator is always singular when the sample size is less than dimensionality of input space, and dimensionality of subspace or the number of these eigenvectors must be less than sample size n. However, with some prior knowledge such as sparsity of correlation or even independence of features, we know that the singular empirical estimator should not be reliable and that is why we turn to Naive Bayes or our method LDAS. However, neither sparsity or independence is necessarily true without any prior knowledge. In this case, it turned out that the data might lie in a lower dimensional space and LDA using the unbiased empirical covariance estimator worked well by finding a reasonable subspace.

Another possible reason might be the redundant features. Recall that in the simulation we assume that the centroids of groups differ from each other in all dimensions equally, which is an ideal case where all the features is informative and valuable as input. Also, in SRBCT data set the 2308 gene features we used were selected among 6567 genes. However, in practice, without feature selection, there must be redundant features which take no information or less information than the noise they have. So in high dimensional space, the first thing we should do is feature selection, which is equivalent to finding a subspace and reducing dimensionality. Traditional LDA somehow is able to implement feature selection as we discussed in the last paragraph while Naive Bayes would take all the features directly assuming all

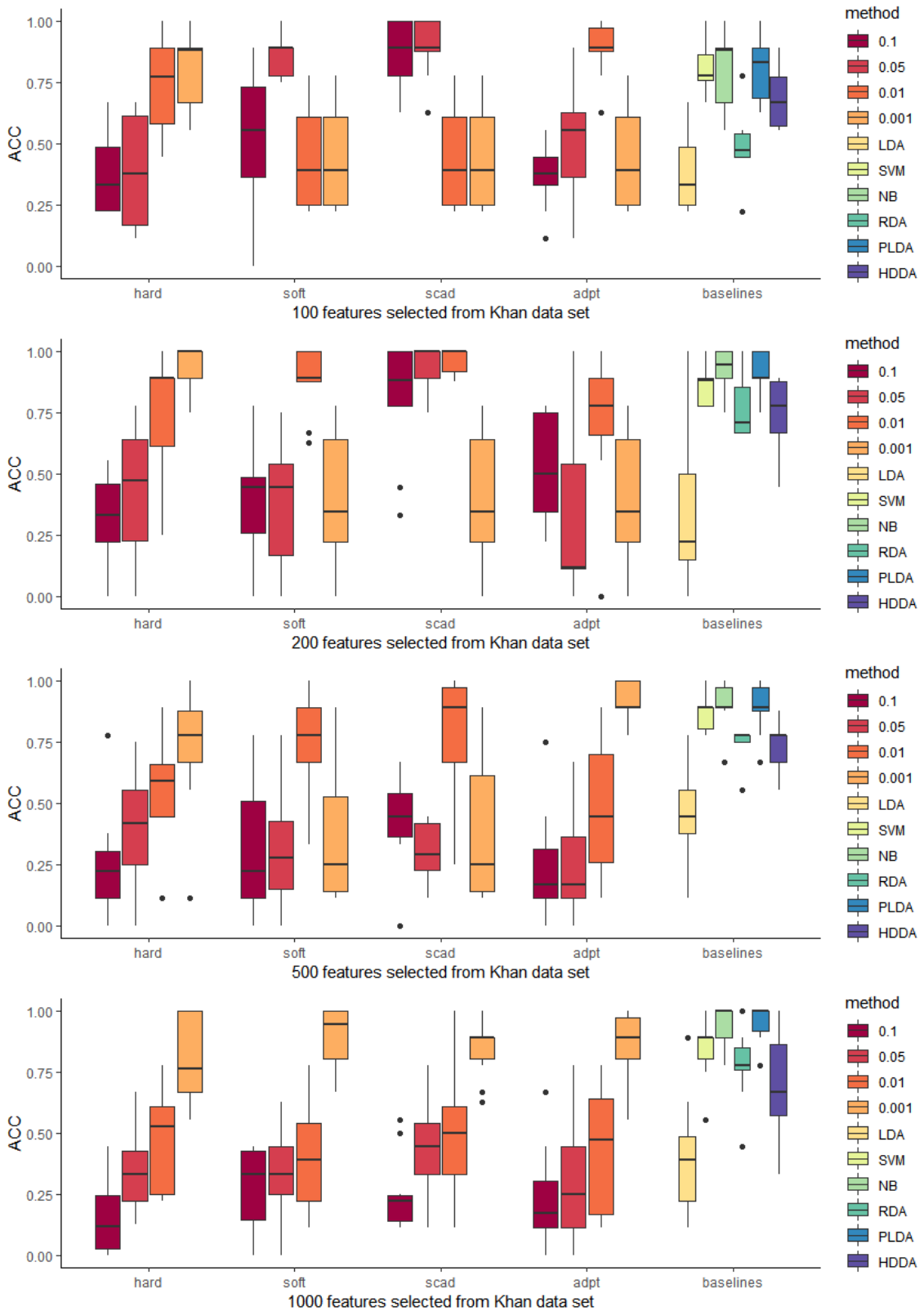

Figure 4.8: 10-fold cross-validation accuracy box-plots of LDAS with different level of false positive rate and thresholding operator compared with other different methods with SRBCT data set

of them are useful. This might explain why Naive Bayes did not work well in this case.

Our approach LDAS also managed to find a balance between LDA and Naive Bayes in this case. As we can see from the box-plots for hard thresholding operator, using different level of false positive rates did not make huge difference, though there seems to be a weak descending trend when 500 and 1000 feature selected. Same as in experiment on SRBCT data set and simulation, hard thresholding operator tend to be the best among four. Especially in this case when simple LDA was the best baseline, hard thresholding operator introduced no bias to large covariance estimations and was the closest one to LDA given large false positive rate.

Table 4.1, which shows 10-folds cross-validation result with 1000 features as an example, provides more detailed information on classification accuracy for each class respectively. Methods such as SVM and HDDA tended to classify samples as group of large size and more supporting evidence can be found after checking the confusion matrices. In other words, these methods are quite sensitive to prior and unable to detect samples from small groups. This is somehow undesired in many situations. For example, here we only have 17 healthy samples while there can be a large number of healthy samples i.e. large control group. If one method is sensitive to prior, treatment group, which is what we usually interested in, can be hardly recognized.

Our approach is not sensitive to prior, which can be considered inherited from its "parents": LDA and Naive Bayes and confirmed by either simulation or real data result. In general, the LDAS performance is competitive to the best baseline LDA in all groups. Except when using false positive rate as 0.001, the numbers of correct classification by LDAS for five groups are approximately same as LDA and due to the small sample size it's hard to tell if one is significantly different from one another.

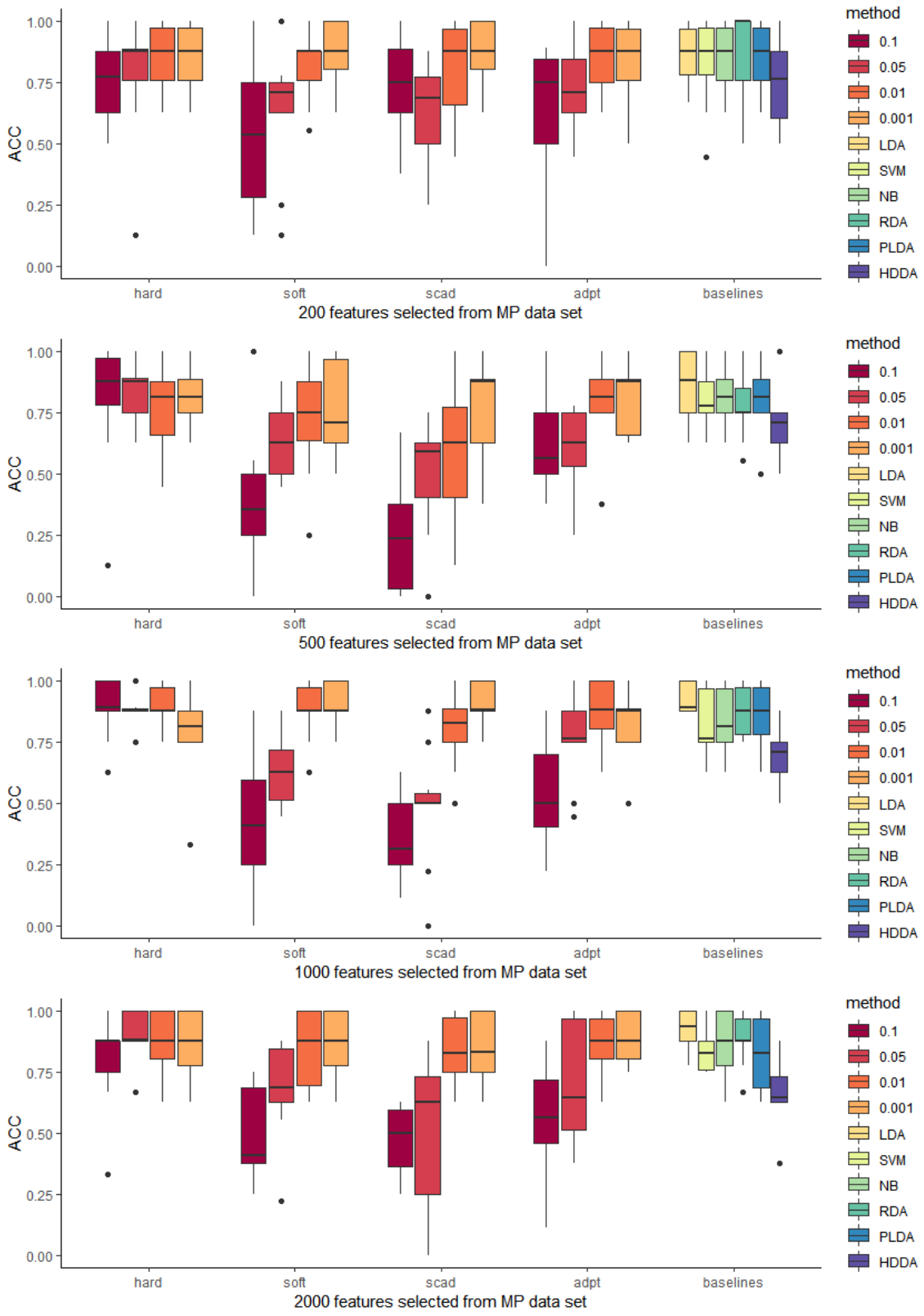

Figure 4.9: 10-fold cross-validation accuracy box-plots of LDAS with different level of false positive rate and thresholding operator compared with other different methods with lung cancer data set

Baselines	AD(40)	CAD(5)	NL(17)	SQ(20)
LDA	37	4	17	18
SVM	**39**	**0**	**0**	**0**
NB	32	4	15	17
RDA	35	5	17	15
PLDA	34	4	16	17
HDDA	36	1	3	16
LDAS	False positive rate = 0.1			
hard	36	4	17	16
soft	17	4	8	5
scad	11	3	8	7
adpt	22	1	10	10
LDAS	False positive rate = 0.05			
hard	34	4	16	19
soft	22	2	13	15
scad	17	3	9	11
adpt	29	4	15	14
LDAS	False positive rate = 0.01			
hard	35	4	16	18
soft	35	3	16	18
scad	30	4	15	17
adpt	35	4	16	16
LDAS	False positive rate = 0.001			
hard	29	4	14	18
soft	36	4	16	18
scad	35	4	16	19
adpt	31	4	15	18

Table 4.1: Number of correct classification for each group given 1000 features in lung cancer data set.

Chapter 5

Conclusion

We started from the basic background introduction on LDA and the limitation of it in high dimensional space, which is the motivation of this thesis. Conventional LDA usually results in low accuracy due to the curse of dimensionality. The performance of LDA relies on the accurate parameter estimation while without abundant samples, there would be large variance for the estimation and, more specifically, singularity issue of covariance matrix estimator when the sample size is less than dimensions. By contrast, Naive Bayes, which can be considered as a simplified LDA under independence assumption, becomes a strong baseline in many high dimensional classification tasks.

Our method LDAS managed to find a balance between LDA and Naive Bayes by sparse assumption and from the numeric result we can conclude LDAS ia able to achieve high accuracy by flexible choice on false positive rate, when the data lies in whether a lower dimensional subspace or not, or in other words, whether the correlation among features are sparse or not. Though not demonstrated by numeric result, we expect in some cases our model would outperform both LDA and Naive Bayes.

We also did a literature review over many other high dimensional classifiers, which can be concluded into two types: modification of LDA under specific assumption and dimension reduction, which includes feature selection and projection. All of them are aiming to reduce the number of parameters and hence, complexity of model. This is consistent with section 2.2, which explained that the small variance is the key factor to low error rate. Our method can be classified as a modification of LDA under the sparsity assumption on covariance matrix. Compared with other two modification SLDA and LDP, our assumption is more weak and general. The other dimension reduction based approaches heavily depend on the covariance estimator which holds huge variance while our method can get better covariance estimation under sparsity assumption.

Bibliography

[1] Peter J Bickel, Elizaveta Levina, et al. Some theory for fisher's linear discriminant function,naive bayes', and some alternatives when there are many more variables than observations. *Bernoulli*, 10(6):989–1010, 2004.

[2] Peter J Bickel, Elizaveta Levina, et al. Covariance regularization by thresholding. *The Annals of Statistics*, 36(6):2577–2604, 2008.

[3] Peter J Bickel, Elizaveta Levina, et al. Regularized estimation of large covariance matrices. *The Annals of Statistics*, 36(1):199–227, 2008.

[4] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[5] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[6] Charles Bouveyron, Stéphane Girard, and Cordelia Schmid. High-dimensional discriminant analysis. *Communications in Statistics—Theory and Methods*, 36(14):2607–2623, 2007.

[7] Tony Cai and Weidong Liu. Adaptive thresholding for sparse covariance matrix estimation. *Journal of the American Statistical Association*, 106(494):672–684, 2011.

[8] Tony Cai and Weidong Liu. A direct estimation approach to sparse linear discriminant analysis. *Journal of the American statistical association*, 106(496):1566–1577, 2011.

[9] Raymond B Cattell. The scree test for the number of factors. *Multivariate behavioral research*, 1(2):245–276, 1966.

[10] Lei Chen. *Curse of Dimensionality*, pages 545–546. Springer US, Boston, MA, 2009.

[11] Line Clemmensen, Trevor Hastie, Daniela Witten, and Bjarne Ersbøll. Sparse discriminant analysis. *Technometrics*, 53(4):406–413, 2011.

[12] Sandrine Dudoit, Jane Fridlyand, and Terence P Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American statistical association*, 97(457):77–87, 2002.

[13] Jianqing Fan and Yingying Fan. High dimensional classification using features annealed independence rules. *Annals of statistics*, 36(6):2605, 2008.

[14] Jianqing Fan, Yingying Fan, and Yichao Wu. High-dimensional classification. In *High-dimensional data analysis*, pages 3–37. World Scientific, 2011.

[15] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

[16] Jerome H Friedman. Regularized discriminant analysis. *Journal of the American statistical association*, 84(405):165–175, 1989.

[17] Jerome H Friedman. On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1(1):55–77, 1997.

[18] Yaqian Guo, Trevor Hastie, and Robert Tibshirani. Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, 8(1):86–100, 2006.

[19] Trevor Hastie, Andreas Buja, and Robert Tibshirani. Penalized discriminant analysis. *The Annals of Statistics*, pages 73–102, 1995.

[20] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning: data mining, inference, and prediction, springer series in statistics, 2009.

[21] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220, 2008.

[22] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.

[23] Richard Arnold Johnson, Dean W Wichern, et al. *Applied multivariate statistical analysis*, volume 5. Prentice hall Upper Saddle River, NJ, 2002.

[24] Adam B Kashlak and Linglong Kong. Nonasymptotic estimation and support recovery for high dimensional sparse covariance matrices. *arXiv preprint arXiv:1705.02679*, 2017.

[25] Javed Khan, Jun S Wei, Markus Ringner, Lao H Saal, Marc Ladanyi, Frank Westermann, Frank Berthold, Manfred Schwab, Cristina R Antonescu, Carsten Peterson, et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature medicine*, 7(6):673, 2001.

[26] Selcuk Korkmaz, Dincer Goksuluk, and Gokmen Zararsiz. Mvn: An r package for assessing multivariate normality. *The R Journal*, 6(2):151–162, 2014.

[27] WJ Krzanowski, Philip Jonathan, WV McCarthy, and MR Thomas. Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 44(1):101–115, 1995.

[28] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[29] Sijin Liu, Xiaotong Shen, and Wing Hung Wong. Computational developments of ψ-learning. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 1–11. SIAM, 2005.

[30] Adam J Rothman, Elizaveta Levina, and Ji Zhu. Generalized thresholding of large covariance matrices. *Journal of the American Statistical Association*, 104(485):177–186, 2009.

[31] Jun Shao, Yazhen Wang, Xinwei Deng, Sijian Wang, et al. Sparse linear discriminant analysis by thresholding for high dimensional data. *The Annals of statistics*, 39(2):1241–1265, 2011.

[32] Pablo Tamayo, Daniel Scanfeld, Benjamin L Ebert, Michael A Gillette, Charles WM Roberts, and Jill P Mesirov. Metagene projection for cross-platform, cross-species characterization of global transcriptional states. *Proceedings of the National Academy of Sciences*, 104(14):5959–5964, 2007.

[33] Jurjen Duintjer Tebbens and Pavel Schlesinger. Improving implementation of linear discriminant analysis for the high dimension/small sample size problem. *Computational Statistics & Data Analysis*, 52(1):423–437, 2007.

[34] Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99(10):6567–6572, 2002.

[35] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[36] DM Titterington, GD Murray, LS Murray, DJ Spiegelhalter, AM Skene, JDF Habbema, and GJ Gelpke. Comparison of discrimination techniques applied to a complex data set of head injured patients. *Journal of the Royal Statistical Society: Series A (General)*, 144(2):145–161, 1981.

[37] WN Vernables and BD Ripley. Modern applied statistics with s, 2002.

[38] Daniela M Witten and Robert Tibshirani. Penalized classification using fisher's linear discriminant. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(5):753–772, 2011.

[39] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

Appendices

Concentration inequality

Recall that given n observations, the empirical estimator is $\hat{\Sigma}^{emp}_{MLE} = \frac{1}{n}\sum_{i=1}^{n}(X_i - \hat{\mu}_k)(X_i - \hat{\mu}_k)^T$ or $\hat{\Sigma}^{emp} = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \mu_k)(X_i - \mu_k)^T = \frac{n}{n-1}\hat{\Sigma}_{MLE}$, which is the unbiased estimator adjusted from MLE. By concentration inequalities, we have that

$$P(d(\Sigma, \hat{\Sigma}^{emp}) \geq Ed(\Sigma, \hat{\Sigma}^{emp}) + r) \leq e^{-\psi(r)}$$

Given the confidence level $1-\alpha$, we can find r_α such that $e^{-\psi(r_\alpha)}$, then we will have the confidence set using empirical estimator

$$\mathscr{C}^{emp} = \{\Sigma \in \mathscr{R}^{p\times p} : d(\Sigma, \hat{\Sigma}^{emp}) \leq Ed(\Sigma, \hat{\Sigma}^{emp}) + r_\alpha\}$$
$$P(\Sigma \in \mathscr{C}_{emp}) \geq 1 - \alpha$$

Now assume we have another estimator $\hat{\Sigma}'$. We want this new estimate to be close to the empirical estimator in the sense of the above confidence set and therefore we choose a $\hat{\Sigma}'$ such that $d(\hat{\Sigma}', \hat{\Sigma}^{emp}) \leq r_\alpha$. Consequently, we have that

$$\begin{aligned}
&P(d(\hat{\Sigma}', \Sigma) \geq Ed(\Sigma, \hat{\Sigma}^{emp}) + 2r_\alpha) \\
&\quad \leq P(d(\hat{\Sigma}', \hat{\Sigma}^{emp}) + d(\hat{\Sigma}^{emp}, \Sigma) \geq Ed(\Sigma, \hat{\Sigma}^{emp}) + 2r_\alpha) \\
&\quad \leq P(d(\Sigma, \hat{\Sigma}^{emp}) \geq Ed(\Sigma, \hat{\Sigma}^{emp}) + r_\alpha) \\
&\quad \leq e^{-\psi(r_\alpha)} = \alpha
\end{aligned}$$

Hence, we can have a confidence set by using $\hat{\Sigma}'$ as well

$$\mathscr{C}' = \{\Sigma \in \mathscr{R}^{p\times p} : d(\Sigma, \hat{\Sigma}') \leq Ed(\Sigma, \hat{\Sigma}^{emp}) + 2r_\alpha\}$$
$$P(\Sigma \in \mathscr{C}') \geq 1 - \alpha$$

In summary, we can build a confidence set for covariance matrix by using another estimator as long as this new estimator is close enough to the empirical estimator.

Key theory for sparse estimator procedure

Given a false positive rate $\eta \geq 0.5$, we can set $\lambda = M_\eta = quantile\{\sigma_{i,j},\ i < j\}$, which removes $100(1-\eta)\%$ of the off diagonal entries, so that a false positive rate of approximately η can be achieved due to the following lemma.

Lemma 5.0.1. *lemma from [24] Let $\Sigma \in \mathscr{U}(k,\sigma), \eta \in [0.5,1), M_\eta$ be the η quantile$\{\sigma_{i,j},\ i < j\}$. Let $\hat{\Sigma}_\eta^{emp}$ denote the corresponding threshold estimator $s_{M_\eta}(\hat{\Sigma}^{emp})$ with i,jth entry denoted as $\hat{\sigma}_{i,j}^{(\eta)}$. Then*

$$\rho(\hat{\Sigma}_\eta^{emp}) = \frac{\#\{\hat{\sigma}_{i,j}^{(\eta)} \neq 0 | \sigma_{i,j} = 0, i < j\}}{p(p-1)/2} \xrightarrow{a.s.} \eta$$

as $d \to \infty$ as long as $k = O(p^\nu)$ for $\nu < 1$

This lemma cannot be extended to arbitrary quantiles or false positive rate. However, the following theorem make it possible to get some properties of a thresholding estimator achieving any desired false positive rate.

Theorem 5.0.2. *theorem from [24] Let $\Sigma \in \mathscr{U}(k,\sigma)$ with $k = O(p^\nu)$ for $\nu < 1/2$. Given a desired false positive rate ρ in$(0,0.5]$, there exist some η such that $\eta = \rho 2^a,\ a \in \mathscr{Z}^+$. Let $\hat{\Sigma}_\rho^{emp}$ denote the hard threshold empirical estimator that achieves a false positive rate of ρ. Then,*

$$|\eta \frac{\|\hat{\Sigma}_\rho^{emp} - \Sigma\|_\infty}{\|\hat{\Sigma}_\eta^{emp} - \Sigma\|_\infty} - \rho| \leq K_1 n \rho^{1/2} p^{-1/4} + K_2 n \rho^{1/4} p^{-1/2} + o(np^{-1/2})$$

where K_1, K_2 are universal constants.

The upper bound on the left side of the inequality above is approximately zero when $p \gg n$. Hence, by this theorem, we can approximately estimate the distance $|\hat{\Sigma}_\rho^{emp} - \Sigma\|_\infty$ with $|\hat{\Sigma}_\eta^{emp} - \Sigma\|_\infty$, where $\hat{\Sigma}_\eta^{emp}$ can be obtained by lemma 5.0.1.